

Copyright

by

Kimberley M. Davis

2000

**Object-Oriented Modeling of Rivers and Watersheds in
Geographic Information Systems**

by

Kimberley M. Davis, B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2000

**Object-Oriented Modeling of Rivers and Watersheds in
Geographic Information Systems**

**Approved by
Supervising Committee:**

David R. Maidment

Howard M. Liljestrand

Francisco Olivera

Dedicated to Momma, Daddy, Joy, and Becca

Acknowledgements

The author wishes to express her gratitude to the numerous people who helped make this work possible. Thanks go to Dr. David Maidment of the Center for Research in Water Resources (CRWR) for his support, guidance, enthusiasm, and patience. Thanks also go to Scott Morehouse, Dale Honeycutt, Steve Kopp, and David Arctur of ESRI California for their assistance and guidance. Special thanks go to Evan Brinton, Julio Andrade, and Brian Goldin of ESRI California for teaching me with patience and good humor. I would like to express my deep gratitude to the members of Dr. Maidment's research group at CRWR, especially Tim Whiteaker, who has been a fantastic co-worker and friend. Thanks to Vicki Samuels for her help with the Figures. Katherine Osborne and Mary Lear—you know why you're here. Also, my thanks to Fiz and Teek for their unexpected and generous assistance. Finally, none of this research would have been possible without the generosity of Jack Dangermond of ESRI California.

August 16, 2000

Abstract

Object-Oriented Modeling of Rivers and Watersheds in Geographic Information Systems

Kimberley M. Davis, M.S.

The University of Texas at Austin, 2000

Supervisor: David R. Maidment

It is possible to take advantage of object-oriented programming and modeling techniques to build a data model of custom features for GIS that supports hydrography and hydrology. The unifying structure that ties river maps to river analysis is the network built from rivers themselves. This thesis describes the design process for the ArcGIS Hydro data model, discusses object-oriented programming concepts, UML, and CASE tools, and shows how to apply the ArcGIS Hydro data model using data from the National Hydrography Dataset. The result is a data structure, not a process-based engineering model, which can be used as-is for most situations, and extended to represent unique conditions if needed.

Table of Contents

List of Tables	x
List of Figures.....	xi
List of Figures.....	xi
Chapter 1 : Introduction	1
1.1 Motivation	2
1.2 Objective and Scope	3
1.3 Geographic Region of Focus.....	4
1.4 Thesis Outline	5
Chapter 2 : Literature Review	7
2.1 National Hydrography Dataset.....	7
2.2 City of Austin.....	11
2.3 Wisconsin Department of Natural Resources	12
2.4 Oregon and Washington Framework.....	13
2.5 British Columbia Watershed Atlas	14
2.6 Hydrologic Engineering Center-Hydrologic Modeling System	15
2.7 Danish Hydraulic Institute	16
2.8 Conclusions	17
Chapter 3 : Methodology.....	19
3.1 GeoDatabase Model	19
3.1.1 Software Overview	22
3.2 Object-Oriented Modeling	24
3.2.1 Definitions	25
3.2.2 Properties.....	26
3.2.3 Unified Modeling Language (UML)	29
3.2.4 Computer Aided Software Engineering (CASE) Tools	30

3.2.5 Component Object Modeling (COM)	31
3.3 Networks	32
3.3.1 Components	32
3.3.2 Conceptual Framework for Networks	36
3.3.3 Linear Referencing	38
3.4 ArcGIS Hydro data model	42
3.4.1 Hydro Network	42
3.4.1.1 Hydro Edges	43
3.4.1.2 Waterbodies	48
3.4.1.3 Hydro Junctions	50
3.4.1.4 Hydro Events	54
3.4.1.5 Catchments and Watersheds	57
3.4.1.6 Hydrologic Response Units	58
3.4.2 Hydro Features	60
3.4.2.1 Hydro Points	63
3.4.2.2 Hydro Lines	68
3.4.2.3 Hydro Areas	71
3.4.2.4 Use of subtypes and subclasses	72
3.4.3 Channel Features	73
3.4.4 Time Series	76
3.4.5 Relationships among objects	76
Chapter 4 : Procedure of Application	81
4.1 Recursive Model Design	81
4.1.1 Summer Internship	81
4.1.2 Work with Consortium	82
4.1.3 Work with ESRI	83
4.2 Implementation in Visio	83
4.2.1 Export to Repository	84
4.3 Generate Code	85

4.3.1 .DLL File	86
4.4 Using the ArcGIS Hydro data model.....	87
4.5 Load Data.....	87
4.5.1 Adding the Object Loader to the Toolbar	89
4.5.2 Use of ArcMap to load data	90
4.6 Apply Schema to data in ArcCatalog	92
4.6.1 Adding the Schema Generation Wizard to the Toolbar	94
4.7 Generate Network.....	95
Chapter 5 : Results	102
5.1 Converting NHD into a Hydro Network.....	103
5.2 Analysis Capabilities	105
Chapter 6 Conclusions	110
6.1 Future Work	113
Appendix A	116
Bibliography	121
Vita	123

List of Tables

Table 2.1	Feature correspondence in the NHD and City of Austin datasets	12
Table 3.1	Spatial Features in ESRI Software.	21

List of Figures

Figure 2.1	Natural and Artificial Reaches on Lewisville Lake, Trinity River Basin.	10
Figure 3.1	The ArcCatalog GUI displaying a preview of a feature class.....	22
Figure 3.2	The ArcMap GUI displaying a set of feature classes	23
Figure 3.3	Examples of Polymorphism, Inheritance, and Encapsulation.....	27
Figure 3.4	The ArcObject set showing the derivation of Simple and Complex Edge and Junction objects.....	33
Figure 3.5	Example of Simple and Complex Edges	35
Figure 3.6	Diagram of the geometric network and logical network tables (Zeiler, 1999, p.132)	37
Figure 3.7	Construction of the Rch_Code for linear referencing.....	39
Figure 3.8	Construction of the LLID for linear referencing	40
Figure 3.9	An illustration of the differences between Relative and Absolute Addressing.....	41
Figure 3.10	Examples of the three subtypes of Hydro Edge	43
Figure 3.11	UML diagram for Hydro Edge.....	46
Figure 3.12	Use of the LinearRef_ID to link Flow Edges.....	47
Figure 3.13	An example of why Hydro Edge derives from Simple Edge.....	48
Figure 3.14	The UML Diagram for Waterbody.....	49
Figure 3.15	An example of Closure Lines separating Waterbodies.....	50
Figure 3.16	The UML Diagram for Hydro Junction	52

Figure 3.17	An example of the three main subtypes of Hydro Junction and ESRI Generic Junction.....	53
Figure 3.18	Examples of Hydro Point Events on Benbrook Lake in the Lower West Fork of the Trinity River	54
Figure 3.19	The UML diagram for HydroEvents	55
Figure 3.20	An example of a linear event near the mouth of the Trinity River represented in the table by the boxed fields and in the map by a red line.	56
Figure 3.21	UML diagram of Watershed, Catchment and Hydrologic Response Unit.....	58
Figure 3.22	Hydrologic Response Units across the city of Austin representing landuse zones.....	60
Figure 3.23	Hydro Features from the National Hydrography Dataset (NHD)	62
Figure 3.24	Aerial photography and derived stream network	63
Figure 3.25	Hydro Points on the Lower West Fork of the Trinity River	64
Figure 3.26	Features that are represented by Structure Points (LCRA, 2000) and (Water CPI, 2000)	65
Figure 3.27	Features that are represented by Flow Change Points (Oregon Water Resources Department, 2000) and (Engineering Review, 1998)	66
Figure 3.28	Features that are represented by Monitoring Points (Conant Custom Brass, Inc., 1999) and (Institut für Meereskunde, 1997)	67
Figure 3.29	Hydro Lines taken from the NHD	69

Figure 3.30	Closure Lines on Corpus Christi Bay	70
Figure 3.31	Hydro Area representing the Inundation Area of Benbrook Lake in the Trinity River Basin.....	71
Figure 3.32	UML diagram of Hydro Features	72
Figure 3.33	The UML diagram for Cross Section and Profile Line	74
Figure 3.34	Illustration of the relationships between Edge Catchments and Hydro Edges.....	77
Figure 3.35	Relating points to time series of hydrologic data	78
Figure 3.36	UML diagram of connectivity rules	80
Figure 4.1	A .dll file about to be registered with RegCat.exe	87
Figure 4.2	Sample of a dataset from the NHD for the Lower West Fork of the Trinity River	88
Figure 4.3	The ArcMap menu for customizing toolbars	89
Figure 4.4	The toolbar customization dialog box in ArcMap.....	90
Figure 4.5	The opening dialog box of the Schema Creation Wizard	93
Figure 4.6	The ArcCatalog menu for customizing toolbars	94
Figure 4.7	The toolbar customization dialog box in ArcCatalog.....	95
Figure 4.8	Preparing the data to generate a network.....	96
Figure 4.9	The Geometric Network wizard is invoked from ArcCatalog	96
Figure 4.10	The results of snapping	97
Figure 4.11	Snap Features dialog box from the Geometric Network Wizard	98
Figure 4.12	Dialog box for enabling Sources and Sinks in the Hydro Network	98

Figure 4.13	Property Inspector window showing the AncillaryRole list for a selected Junction.....	99
Figure 4.14	Using the flow direction information just recorded, measure values can be assigned to the Edges in the network in ArcInfo version 8.1.	101
Figure 5.1	The Lower West Fork of the Trinity River in ArcGIS Hydro data model	102
Figure 5.3	The four types of Network Flags.....	107
Figure 5.4	Different results configurations from the same upstream trace task.....	108

Chapter 1: Introduction

A GIS is a Geographic Information System, a software package for creating, viewing, and analyzing geographic information or spatial data. GIS is a class of software, just as word processors or databases are. GIS was originally developed and used only to create maps. As the software has evolved over the last thirty years, its cartographic capabilities have been augmented by analysis tools. Using a GIS, maps displaying spatial data can be analyzed to discover “why things are where they are and how they are related” (Mitchell 1999, p. 10). Such analyses can be used in decision support, emergency management, planning, maintenance, and many other applications. ESRI is a leading company in the GIS industry and strongly supports research and development, both in house and in cooperation with users of their software. The latest versions of their ArcInfo and ArcView software are collectively known as ArcGIS and result from the years of creative development and research ESRI has conducted.

ArcInfo 8, the latest GIS software created by ESRI, incorporates new ideas in computer science that allow spatial data to be handled in a whole new way. Advances in database software and computer hardware have made possible a new type of GIS. This new software makes use of Object-Oriented Modeling and Programming techniques and of database technology that allows large binary objects (i.e., files) to be stored in tables inside a relational database. The ability to store files in tables makes it possible to store the coordinates of spatial features as files in the same table with the relational data for the feature. Object-Oriented

Programming is a technique that enabled the development of the windowed computer environment. It is the opposite of Procedure Oriented Programming, which was the basis for menu-driven programs that prompted the user for input and followed a linear algorithm, usually represented by a flowchart. Object-Oriented Programs do not have that type of linear structure. Instead, they make objects available and wait for the user to interact with one, which invokes a piece of the program in response. To take advantage of the new object-oriented construction of the software, users need to develop data models that work with it. Since the development of ArcInfo 8.0, ESRI has been cooperating in the creation of data models for use by various user groups in GIS. To that end, work began in the summer of 1999 on the hydrology and hydrography data model, called the ArcGIS Hydro data model. ESRI and the Center for Research in Water Resources (CRWR) at University of Texas founded the GIS in Water Resources Consortium in September of 1999. The Consortium is composed of members from industry and government at the national, state and local levels and was founded to develop applications for GIS in Water Resources. The primary goal of this development task is to create the ArcGIS Hydro data model for representing rivers and watersheds in GIS. The creation process and the specifics of the ArcGIS Hydro data model are the subject of this work.

1.1 MOTIVATION

Rivers and watersheds have long been modeled by engineers. They have long been mapped by cartographers. Rarely have these two elements been coupled together in order to take advantage of the spatial analysis built into GIS

programs and the hydrologic and hydraulic analyses available in engineering models. A purpose of the ArcGIS Hydro data model is to use GIS to facilitate creation of spatial data for hydrologic and hydraulic models such as HEC-HMS, HEC-RAS, MIKE 11, and MIKE-SHE.

The landscape represented by this model is complex. It contains rivers and lakes, natural and man-made watercourses, large expanses of land, and selected locations at which scientists seek to monitor water flow and quality. This set of spatial features and time series data about them are the fundamental objects from which hydrologic and hydraulic engineering analyses are performed. The task of water resources engineers is to mitigate the damage caused by floods, droughts, and pollution. Computer models are used to best predict the danger and deal with it. These models, however, are only as good as the data available to them. The ArcGIS Hydro data model can be used to automate spatial data development and to make the data more relevant to the analysis tasks it must support.

1.2 OBJECTIVE AND SCOPE

There are many companies and software packages within the GIS industry. This research has been supported by ESRI, an industry leader, and is performed within the framework of ESRI's ArcGIS software. This prototype model is tailored for use with ArcGIS, and has been designed to take advantage of the functionality available within ArcGIS.

The ArcGIS Hydro data model is not intended to be a hydrologic model. It is not programmed with the necessary engineering equations to perform

hydrologic simulations or analysis. It is simply a data model, a framework for storing hydrographic and time series data in a database so that it can be transported into hydrologic modeling packages. The essence of a data model is a database schema, an arrangement of related tables with particular fields for storing particular data. Data models can be used to facilitate data exchange by ensuring that certain minimum standards of compatibility are met. This model is intended to cover many of the situations found in hydrology and hydrography as it exists now and to be extensible for use in special situations.

The data model works with ArcGIS to take advantage of the new Utility Network Analysis capabilities and the integration of raster and vector data allowed in this new software. These tools allow for a greater degree of automation and more extensive analysis capabilities. Still, the model is for data organization only and is not meant to supplant process-based engineering models.

1.3 GEOGRAPHIC REGION OF FOCUS

The ArcGIS Hydro data model was constructed with the intent that it be useful the world over. It is compatible with the hydrology datasets being built in the United States, but it is not limited to such datasets. For the purposes of this project, the Lower West Fork of the Trinity River in Texas was used as an example of how to apply the data model to existing hydrographic data. The data were provided by the NHD.

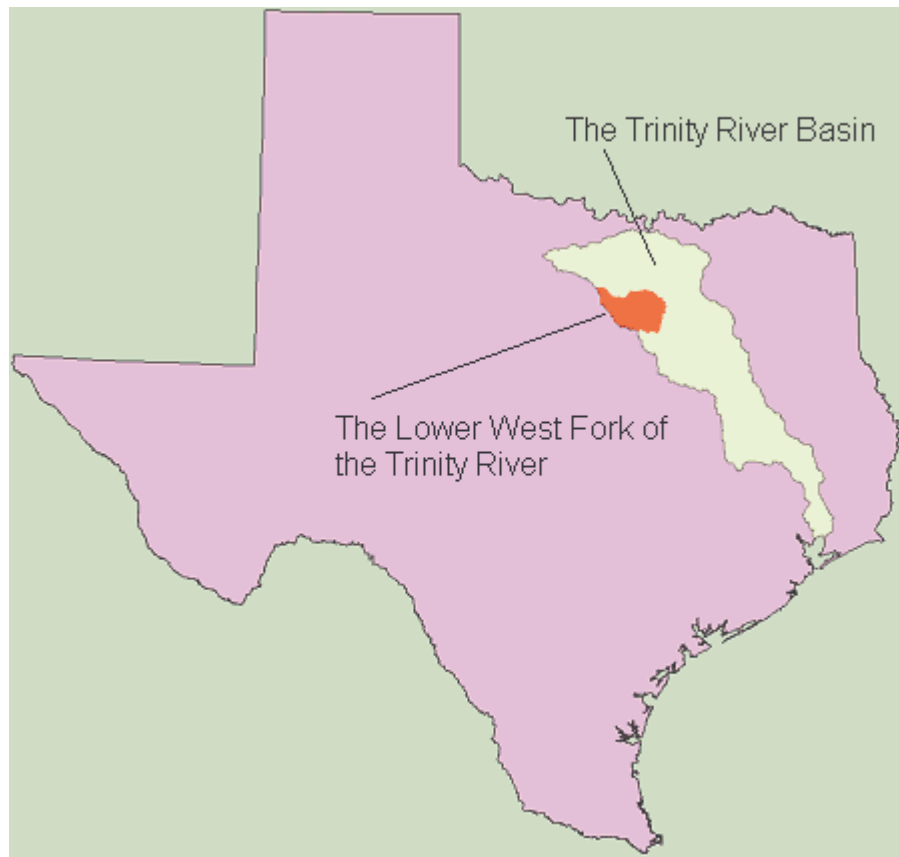


Figure 1.1 The Lower West Fork of the Trinity River in Texas

1.4 THESIS OUTLINE

This paper is divided into six chapters. The first provides a broad overview of the objective, scope, and motivation for the development of the ArcGIS Hydro data model. The second reviews the other data models and literature which were studied in the creation of this model. The third is an introduction to the theory and methodology which made this project possible. Most of the background information presented in this chapter may be unfamiliar to hydrologists and hydrographers, but it is crucial to understanding the decisions

made in the model design. The model itself, its components and their attributes and relationships, are also detailed in this chapter. The fourth details the steps taken to apply the procedure, which consists of two phases. The first phase is creation of the data model, the second is use of the model with the ArcGIS software. The fifth is an example application illustrating how the data model can be used. The sixth draws conclusions and points to future work that may be undertaken.

Chapter 2: Literature Review

Data models from national, regional, local and international sources have been studied during the process of developing the ArcGIS Hydro data model. The insight gained from them has been invaluable and they have been a yardstick by which to measure the success of this new model in covering the water resources domain. The National Hydrography Dataset (NHD) and the Hydrologic Modeling System (HMS) from the US Army Corps of Engineers Hydrologic Engineering Center (HEC) have been foremost among the studied models.

2.1 NATIONAL HYDROGRAPHY DATASET

The National Hydrography Dataset (NHD) was developed by the United States Geological Survey (USGS) and the EPA in order to provide a standardized national set of data for hydrography, i.e., the mapping of rivers. It is both a data model and a dataset organized on hydrographic bases. This dataset was built from USGS topographic maps at 1:100,000 scale and is an inventory of the hydrographic features found on those maps. There are 52 separate feature classes in the NHD. Most of the features are water bodies or rivers, but a few features are landmarks of various kinds or man-made water control structures. Five types of features make up the stream network in NHD, Stream/River, Canal/Ditch, Pipeline, Artificial Path, and Connector. The first four feature types in the preceding list account for 91% of the features included in NHD datasets nationwide (Maidment 1999, p. 4). The challenge of working with the NHD was ensuring that all 52 of these feature types could be represented in the ArcGIS

Hydro data model without sacrificing simplicity. The challenge was met by keeping the ArcGIS Hydro data model small and creating base classes from which to derive each type of hydrographic feature. The properties of the features in the ArcGIS Hydro data model are meant to be modified. The feature classes themselves, as well as their attributes, are prepared for editing and customization by the user, allowing the core model to be fairly basic while giving the user a good framework from which to begin extension.

Data in the NHD is organized into three levels. At the lowest level are the points, lines (arcs), and polygons that represent every feature in the NHD. If the lines and points of the source maps were gathered up without regard for feature type, this base set of features would result. They are the undifferentiated raw material from which the rest of the NHD is built. The set of base features is stored in a single ARC/INFO coverage called NHD. From these base features, aggregations and concatenations can be used to build meaningful collections of features, such as the drainage network. The lines which represent Reaches (portions of the drainage network and marine coastlines) are collected into a route subclass called Route.rch. The members of Route.rch that participate in the drainage network are collected together as a route subclass called route.drain. A route subclass is simply a series of lines, like bus routes, which are wholly derived from an underlying network of lines, like a city street map. The same is done with the polygons so that polygons representing waterbodies are collected into a region subclass called Region.wb, and those which are significant enough to be assigned reach codes are gathered into Region.rch. There is another set of

subclasses which is used to store landmark data that is not part of the drainage network or the waterbodies, these are Route.lm and Region.lm. They contain additional information from the maps which serves as landmarks, but which does not bear on the drainage network.

The structure of the ArcGIS Hydro data model borrowed the concepts of Artificial and Coastline Reaches from the NHD. Coastline Reaches are used to delineate ocean or sea shorelines. They support drainage areas in which runoff flows overland into a waterbody without being gathered into stream. Artificial Reaches are used in the NHD to represent centerlines of water bodies. Single-line geometry for all watercourses and water bodies is necessary to enable the use of network tracing algorithms. The single lines down the center of the reservoir arms in Lewisville Lake, shown in Figure 2.1, were developed by the USGS for the NHD.

In Figure 2.1 Natural Reaches (blue lines) are shown flowing to the border of the lake, which is represented by the purple area. At the shoreline of the lake, the Artificial Path Reaches (black lines) begin to trace the path of flow through the lake to the dam, where the Natural Reach begins again. This allows network tracing algorithms to continue to operate when the network lines run into areas like lakes and swamps. Without the Artificial Paths, a trace algorithm would not be able to process the lake at all.

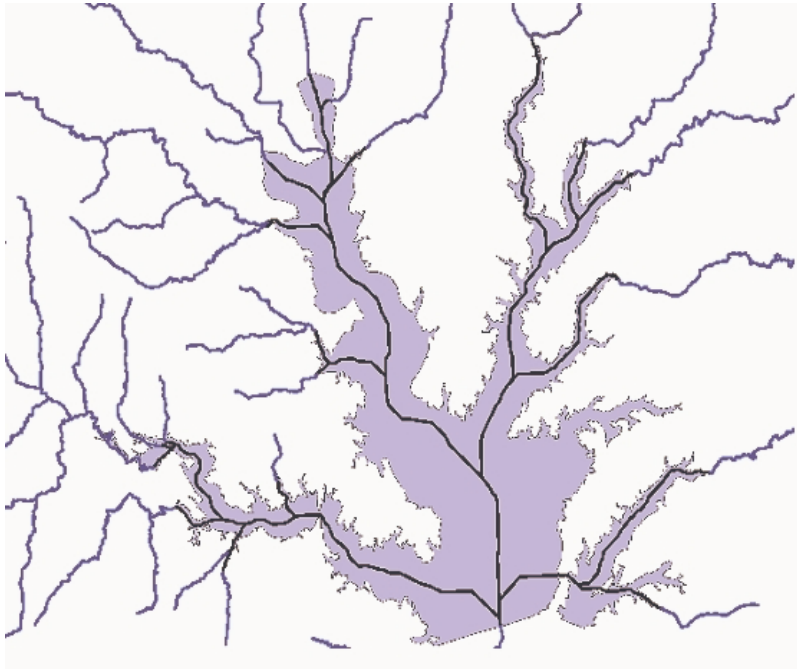


Figure 2.1 Natural and Artificial Reaches on Lewisville Lake, Trinity River Basin.

Arcs in the NHD point downstream; they are digitized so that each arc carries flow toward the basin outlet. In cases of divergent flow, and particularly in the case of loops where flow diverges and then converges again, the network flow table helps determine flow direction. The network flow table, or the flow validation table, is a tool for tracing through a tabular version of the spatial network. For every reach-to-reach connection there is an entry in the table listing the “flows from” reach and the “flows to” reach. The flow validation table also contains information about “underpasses”, the rare instances in which waters cross one another, but do not converge, such as when an aqueduct carries water above a stream. NHD data is currently available for most of the country without

flow validation information. Flow-validated datasets are replacing the non-flow-validated versions as they become available.

2.2 CITY OF AUSTIN

The City of Austin has recently digitized a stream network from aerial photogrammetry. The information is a highly detailed, large-scale data inventory for management of the city stream network. In the course of developing and verifying this network, trends were observed in the data. All of the streams and associated water bodies fit into one of five categories: natural streams, constructed channels, water bodies (with centerlines), pipelines, or connectors. Connectors are places where, although no water feature appeared in the aerial photographs, knowledge of the network and site inspections indicated such a feature was there. The simplest example of a connector is a short line connecting the two parts of a stream on either side of a bridge that obscured the stream in the aerial photograph. These five types of lines, distinguished from one another on a cartographic basis, also reflect the network line types found in the NHD. The correspondence between these two models is shown in Table 2.1. The ArcGIS Hydro data model was constructed to be able to represent all these types of features in behavioral categories, rather than cartographic ones. That is, rather than creating classes for all of the various types of channels, classes were created for the network, representing the behaviors of the features. There are no natural or constructed channels in the ArcGIS Hydro data model; if a channel carries Flow, it is a Flow Edge regardless of its physical representation. However, because artificial paths behave differently, representing flow through bodies of water and requiring quite

a different hydrologic modeling approach, they are a different class in the ArcGIS Hydro data model.

NHD	City of Austin
Stream/River	Natural Stream
Canal/Ditch	Constructed Channel
Pipeline	Pipeline
Connector	Connector
Artificial Path	Centerline

Table 2.1 Feature correspondence in the NHD and City of Austin datasets

2.3 WISCONSIN DEPARTMENT OF NATURAL RESOURCES

The Water Division of the Wisconsin Department of Natural Resources (WiDNR) has begun developing a statewide hydrography dataset at 1:24,000 scale for use in managing water-related data. There are many types of data located on, in, and around water, which have typically been scattered throughout the Water Division of WiDNR. The hydrography dataset will be an inventory model of these data, which will be gathered together and referenced to the appropriate waterbodies and streams in a standard fashion. The features gathered for the hydrography dataset are those which appear on the USGS and United States Forest Service (USFS) 7.5-minute quadrangles. There are many line types which occur in the WiDNR database schema. The most significant concept pulled from the WiDNR data into the ArcGIS Hydro data model was the closure line. The WiDNR hydrography dataset makes use of closure lines to separate

waterbodies from each other. The WiDNR draws distinctions between classes largely on cartographic bases, but this is not strictly the case. Naturally occurring waterbodies in the network are distinguished from man-made waterbodies—a cartographic distinction. In contrast, if a natural stream contains a section which flows through a man-made channel, the whole thing is labeled a natural stream, and the man-made section is not distinguished from the rest—a behavioral grouping. Furthermore, while the WiDNR model contains all the other elements of the drainage network found in the NHD and City of Austin datasets, it excludes the “Connector” data type. In cases where a connector is required to maintain the connectivity of the network, such as where a stream passes through a culvert and disappears from the source map for a short distance, it is simply digitized as part of the stream. The same is true for flow paths that pass through wetlands.

2.4 OREGON AND WASHINGTON FRAMEWORK

The states of Oregon and Washington in the Pacific Northwest have developed a hydrographic data model to facilitate management of surface water data (Washington State Department of Ecology, 1999, p. 1). A characteristic feature of the Oregon and Washington Framework data model is its use of the LLID system to identify water features. The LLID is a system of unique identifiers for river reaches, based on the Longitude and Latitude of the mouth of the river. The LLID applies to a given stream from its mouth to its headwater, which can lead to very long stream segments. While such long reaches are not directly accommodated by the ArcGIS Hydro data model, the shorter reaches

supported by the model can be strung together with attribute values to accomplish the same end.

The Oregon and Washington Framework includes a hydrography data dictionary that incorporates four basic feature layers: Watercourses, Water Bodies, Water Body Shorelines, and Water Points. Features from one of the four layers in the Washington Hydrography Framework are divided into broad behavior categories and are specified further by cartographic codes. There are 13 hydrographic categories which represent a behavioral classification of the features and 36 categories in the cartographic code system. The cartographic codes are used to identify such features as man-made channels, braided channels, natural channels, islands, sandbars, lakes, oceans, glaciers, and reservoirs within the hydrographic categories. Features from a single layer of the Washington Hydrography Framework fall into more than one class in the ArcGIS Hydro data model, based on their combination of hydrologic and cartographic codes.

2.5 BRITISH COLUMBIA WATERSHED ATLAS

In Canada, the British Columbia (BC) Ministry of Environment, Lands and Parks has developed a Watershed Atlas for the province which has as its aim the support of high quality cartographic outputs as well as analysis. The particular focus of the analysis supported by the Watershed Atlas is on the drainage network, which is continuous throughout each watershed. There are 55 feature classes in the Atlas, and most of these are different types of shorelines or network connection lines (BC Ministry of Environment, Lands and Parks, 1996, pp. 11-26). The Watershed Atlas also makes use of nested polygons to describe

the drainage area of a network of streams. This system is similar to the HUC codes that are used to subdivide the NHD.

The BC data model makes use of a network of lines representing streams and centerlines of waterbodies to define the drainage network. The topological rules for capture and coding of features are quite complex in the BC Watershed Atlas, but they support very detailed traces and searches on the network through tabular connections and spatial connectivity. The shorelines, banklines, and river lines of the BC data model are digitized in the upstream direction by convention, allowing directed traces to be performed on the network. In cases of divergence, the water features are attributed as being part of the primary or secondary flow.

One thing the BC Watershed Atlas does differently from other models is to tabularly relate network elements to the watersheds they are located in. The ArcGIS Hydro data model implements such a relationship as well. Creation of this stream-to-landscape relationship allows traces through the stream network that return information about the land through which the network flows. The nested watershed system used in the BC Watershed Atlas gives each feature in the inventory a unique feature code and a unique watershed code. Each feature also has attributes defining what the next downstream drainage element and downstream watershed are.

2.6 HYDROLOGIC ENGINEERING CENTER-HYDROLOGIC MODELING SYSTEM

In the US Army Corps of Engineers Hydrologic Engineering Center's Hydrologic Modeling System (HEC-HMS), there are seven kinds of hydrologic objects, watersheds, river reaches, junctions, reservoirs, diversions, sources and

sinks. This type of model is behavioral in nature, and distinguishes features solely on the basis of how they affect the movement of water over the landscape. Regardless of type, all objects in HEC-HMS have an identification number. Each object in the model knows what the next downstream object in the water flow sequence is, thus a schematic model of the object connections can be drawn. The schematic looks like a skeleton version of the basin and forms the basis of the sequencing of the hydrologic flow computations. This concept of connecting watersheds to the network by a unique identifier is the foundation of the same relationship in the ArcGIS Hydro data model. In addition to carrying this concept into the ArcGIS Hydro data model, care was taken to ensure that each of the objects modeled by HEC-HMS has a representation in ArcGIS Hydro.

2.7 DANISH HYDRAULIC INSTITUTE

The Danish Hydraulic Institute (DHI) is the developer of MIKE 11, a software package used in river and floodplain modeling. MIKE 11 has a GIS-enabled version that allows limited interface with ESRI's ArcView software. One of the goals of the ArcGIS Hydro data model is to improve on the amount and degree of interoperability demonstrated by programs such as MIKE 11 and ArcView.

The data model for MIKE 11 is behavioral in nature and includes a stream network equipped for linear referencing, three dimensional channel topography, and time series data. The channel data can come from a digital terrain model or from survey data, and can support linear events containing data about the roughness of the channel. All of these objects are included in the ArcGIS Hydro

data model. DHI's participation in the Consortium and in data model design meetings helped shape the Catchment and Watershed objects, the Time Series objects, and the terminology used in the model. The insight DHI provided into European systems of river management was invaluable when designing the linear referencing system, as well. The chainage system used in MIKE 11 is analogous to the river mile measures used by some environmental agencies in the United States, but it begins the measures with a 0 value at the upstream end, and the values increase going downstream. Most of the systems used in the US measure up from the mouth of the river or stream in question.

2.8 CONCLUSIONS

The NHD, City of Austin, WiDNR, Oregon and Washington Framework, and BC Watershed Atlas models are all inventories of hydrographic features. The Oregon and Washington Framework uses some behavioral categories to aid in distinguishing its features, but it is still essentially an inventory. HEC-HMS and MIKE 11 are both procedural engineering models that organize the data they work with in terms of its behavior in the procedures. All of the models discussed focus around the stream network and on the utility gained in analysis by tracing through the streams. Only in the BC Watershed Atlas and ArcGIS Hydro data model can watersheds be traced by means of the stream network. In ArcGIS Hydro features are behaviorally distinguished; with more categories than the 7 found in HMS or 3 from MIKE 11, it is able to represent detailed hydrography. The ArcGIS Hydro data model can bridge the gap between inventory models and

behavioral models, thus translating datasets built using inventory data models into a form that behavior-based models can easily use.

Chapter 3: Methodology

The development of ArcGIS was made possible, in part, by advances in hardware and relational database management systems (RDBMS) software that make accessing an RDBMS during spatial operations reasonably quick. Other developments which were incorporated into the design of ArcGIS include object-oriented programming and network theory. Taken together, these advancements allowed the introduction of ArcGIS and the geodatabase model for spatial data management on which it is built. This chapter explains the geodatabase model and explains how it differs from the coverage and shapefile models which preceded it. It also introduces the critical concepts which define Object-Oriented Modeling, and explains network theory as it is used in ArcInfo 8. Finally, this chapter defines the elements of the ArcGIS Hydro data model, their properties, and the relationships among them.

3.1 GEODATABASE MODEL

ARC/INFO through version 7 was based on the coverage model in which points, lines, and polygons (areas) have topology. Topology is the set of relationships among neighboring features that gives them a limited awareness of their surroundings. The practical result of topology is that points know which arcs are connected to them, arcs know which points constitute their origin and destination, and arcs know polygons they form the perimeters of (AGI, 1996, “topologically structured data”) Any time the feature geometry is edited, the topological relationships must be rediscovered and refreshed by the software.

This rigid structure allows efficient spatial queries and tracing on line networks, but is unwieldy for those applications in which such topology is not an advantage. In a coverage, spatial data and attribute data for features are stored in proprietary format files that cannot be used or accessed by other software. The data structure was intended to be as close to a relational database as possible, within cost and performance constraints. Indeed, it appears to perform much as a relational database does, though the internal workings are different.

ArcView was the next major ESRI software package. It made use of a looser spatial data structure, the shapefile model. In ArcView versions 1-3 there are no topological relationships among the features. While still employing separate binary files to store coordinate data for features, it used a standard RDBMS format called dBase to store attribute tables. Taken together, the dBase and binary coordinate files were called shapefiles.

In ArcGIS ArcInfo 8 and ArcView 8, a hybrid of these two spatial data structures is used. The following two classes of features are defined: simple features and network features, as shown in Table 3.1. Simple features in ArcGIS are similar to the point, line, and polygon shapefiles in ArcView. The network features Junction and Edge correspond to Node and Arc in the earlier versions of ARC/INFO, though the network features in ArcInfo 8 incorporate meaningful advances in design over nodes and arcs.

ARC/INFO (Coverage)	ArcView (Shapefile)	ArcInfo 8 (Geodatabase)
	Point	Point—Simple Feature
	Line or PolyLine	Line—Simple Feature
	Polygon	Polygon—Simple Feature
Node		Junction—Network Feature
Arc		Edge—Network Feature
Polygon		N/A

Table 3.1 Spatial Features in ESRI Software.

There are numerous benefits to using the geodatabase model rather than the coverage or shapefile models. One benefit of is that it uses off-the-shelf relational databases like Oracle and Microsoft Access to store the data, allowing spatial information to be easily combined with non-spatial data already entered into such databases. Another benefit of this model is that data entry and editing are more accurate, due to the ability to enforce valid ranges for attribute values with data validation functions. The values which can be stored in a particular attribute field may be restricted to certain integers by using a Coded Value Domain, or to a range of decimal values using a Range Domain. The objects stored in tables in the geodatabase, called geobjects, are more intelligent in their display and behavior than features in ARC/INFO or shapes in ArcView, and conform better to the reality of their existence. The geodatabase also enables relationships among geobjects; relationships allow easier and more accurate interpretation and editing of data. Furthermore, geobjects can be composed of elliptical, circular, and irregular curves. Coverages and shapefiles allowed only straight line segments.

3.1.1 Software Overview

In addition to the changes in the spatial data structure and the name, the software itself has changed significantly from ARC/INFO 7 to ArcGIS. ARC/INFO was a command-line program reminiscent of Microsoft's DOS.

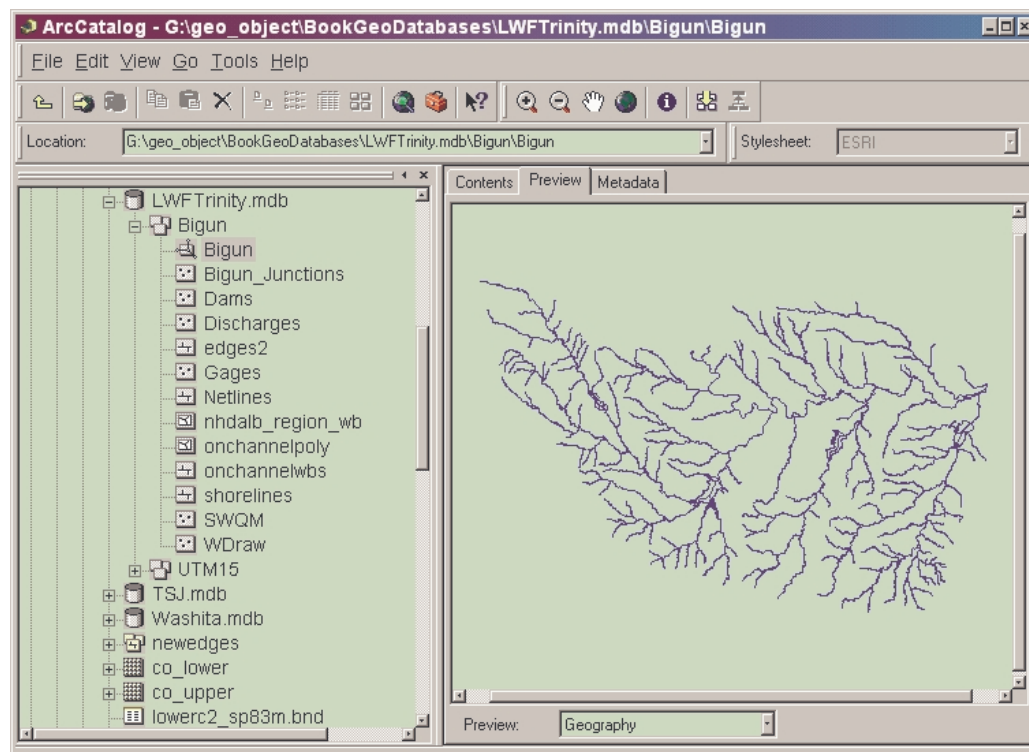


Figure 3.1 The ArcCatalog GUI displaying a preview of a feature class

ArcInfo 8 makes full use of a Graphical User Interface (GUI) designed to make its features more accessible in the Windows-driven software market, and has made great strides in user friendliness and usability. ArcGIS is a suite of software comprising ArcInfo 8 and ArcView 8. The numbering of ArcView versions jumped from 3.2 to 8, to create parallelism with ArcInfo 8 and to

emphasize that ArcView now uses the geodatabase model. The ArcInfo 8 software package consists of three separate components, ArcCatalog, ArcMap, and ArcToolbox, each of which is tied loosely to the others. ArcInfo 8 is properly viewed as an integrated application suite, and ArcView is a lightweight auxiliary component for viewing and querying geodatabases. ArcCatalog is the component used for file, schema, and metadata management; it is shown in Figure 3.1. ArcMap, shown in Figure 3.2, is used for visualizing data, generating maps, performing all spatial analyses, and editing spatial and attribute data.

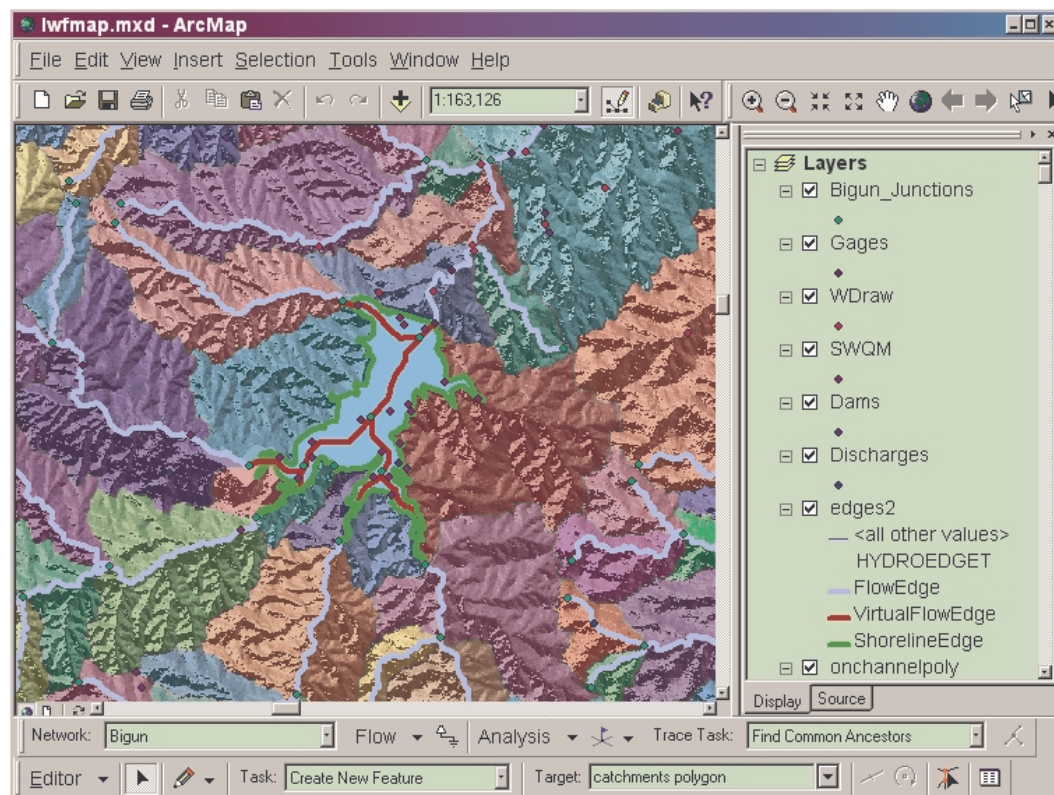


Figure 3.2 The ArcMap GUI displaying a set of feature classes

ArcToolbox is a collection of commands which can be run to perform tasks that do not properly fit into either ArcCatalog or ArcMap, such as format conversion or projection transformation. Some commands and wizards are available in more than one place in the package, such as the Object Loader which runs in both ArcCatalog and ArcMap.

3.2 OBJECT-ORIENTED MODELING

In the past, features stored in an ESRI GIS database were all virtually the same, and very basic: points, lines and polygons. Features of the same type were collected together and assigned other attributes to more fully define their nature. Examples of attributes include length or area of the feature, owner name, agency responsible for maintenance, designer, color, elevation, operational status, etc. With the advent of object-oriented programming in GIS, users are no longer limited to the simple points, lines, and polygons of the past. Lines representing a road, a wall, a shore, or the center of a river are not functionally equivalent. They do not have the same behavior, so they are not expected to respond to operations in the same way. As a simple example, a river line almost always has one-directional flow, while a street usually has two-way flow. A trace algorithm should handle these two types of lines differently, and should not operate on certain line types at all. Before ArcGIS, there was no way to differentiate these different lines; that is no longer the case. Features are still based on one of the three basic geographic representations, but now there is a greater range of operations available. Features have interfaces which define how they respond to requests, and each object's interfaces can be different. This is the basis for

categorizing objects by their behavior, rather than their cartographic representation.

3.2.1 Definitions

Object: (General) an entity with uniquely defined properties, methods, relationships and interfaces that can be processed and manipulated by software

(ArcInfo 8) a Row in a database table that is uniquely identified by its ObjectID

Geoobject: an object that can be stored in a geodatabase

Feature: a kind of Object with a shape that can be represented on-screen

Class: a group of like Objects with the same attribute schema and behavior
(Booch, 1991,p.93)

Abstract Class: a class that is never instantiated and exists solely to consolidate common attributes and behaviors from classes below it in the hierarchy

Concrete Class: a class that can be instantiated, the opposite of abstract

Property: an attribute of an Object (e.g. the 7-day, 10-year low flow is a typical property of the class Flow Edge)

Method: an inherent behavior of an object (e.g. all Features have a method for drawing themselves on-screen)

Polymorphism: the property of objects which allows a request to operate differently on different objects, without requiring that the object class be known in order to make the request

Encapsulation: the property of objects which ensures that all objects are accessible only through clearly defined interfaces so as to protect the internal workings of the object

Inheritance: the means by which all derived objects in lower tiers of a model inherit properties, methods, and behaviors from the basic objects in higher tiers of the model

CASE: Computer Aided Software Engineering, the task of using a computer to automate the development of software

COM: Component Object Modeling, a software development standard devised by Microsoft which enables compliant programs to interact with each other's data through standard interfaces

UML: Unified Modeling Language, a standard graphical language for writing software blueprints, used to visualize, specify, construct and document the artifacts of a software-intensive system (Booch, 1999, p. 13)

DLL: Dynamic Link Library, a file extension denoting a code "library" file that contains the instructions used by the computer to generate objects

3.2.2 Properties

The critical characteristics of object-oriented modeling are polymorphism, encapsulation, and inheritance. Polymorphism means that the same operation may behave differently on different classes. (Rumbaugh, 1991, p. 2). Imagine a base class called shape with two properties defined on it, Area and Perimeter, as seen in Figure 3.3. A standard of the UML is to show each object as a box with three compartments. The top compartment is the class name, the middle one is

the properties of the class, and the lower compartment is for the methods the class implements. As shown in Figure 3.3, three classes derive from shape: circle, square, and triangle. Polymorphism enables the programmer to define different area and perimeter methods for the derived circle, square and triangle classes. No matter what kind of Shape an object is, applying the area or perimeter method to it causes the object to run the calculation stored internally for that request and return the correct results (Webopedia, 1999).

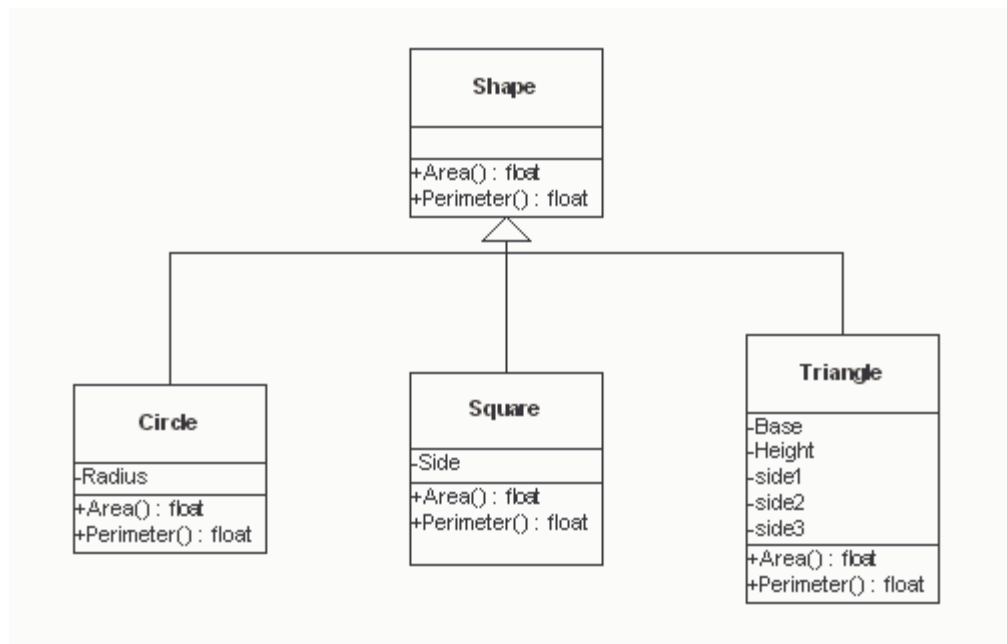


Figure 3.3 Examples of Polymorphism, Inheritance, and Encapsulation

Encapsulation says that an object has crisply defined borders and can only be accessed through certain interfaces provided by the software (Hathaway, 1996, §1.2). This keeps the inconsequential inner workings of the object hidden from users and causes objects to operate in a manner akin to a “black box.” Physically,

an object is a series of code and a collection of variables. Encapsulation protects the object's variables from being inappropriately edited by the user. Because of encapsulation, a developer examining the objects can see that each of the derived classes has a method Perimeter and Area, but not how they are calculated. That information is not editable, so it is not made visible. In Figure 3.3, the formulas for perimeter and area are not visible to the user, although the methods for calculating them are, because of encapsulation.

Properties and methods are passed from one class to another class by inheritance. This allows portions of code to be reused without being compiled for each object that uses them and makes for more concise diagrams of object models. The derived classes Triangle, Circle, and Square inherit the methods Perimeter and Area from the base class Shape. The triangle symbol beneath the Shape class in Figure 3.3 signifies inheritance, and is called a generalization relationship. It means that the classes Triangle, Circle, and Square are kinds of the class Shape. It is not necessary to specify the methods Perimeter and Area when defining the child classes. These will exist for the derived classes, because they inherit them from the base class. For this example, the inherited methods are displayed but that is optional, they are assumed to be present. This is a simple example, so the advantages provided by inheritance are not readily apparent. In a more complex model with hundreds of constituent classes, the savings are considerable in terms of space on the object model diagram, computer memory to store the model, and time needed to interpret the object diagram.

3.2.3 Unified Modeling Language (UML)

UML is a graphical language developed by computer scientists in order to document software systems and aid in design of object-oriented programs. A UML static structure diagram is a software blueprint for object-oriented programs, analogous to the flowcharts used to design menu-driven programs. Over 50 systems for object-oriented software documentation proliferated between 1989 and 1994. In the mid-1990s, three systems began to be recognized as the best and most dominant in the field. Because each of the three had strengths and weaknesses, the authors began to borrow from one another. The Booch, Object-Oriented Software Engineering (OOSE), and Object Modeling Technique (OMT) methods, authored by Grady Booch, Ivar Jacobson, and James Rumbaugh, respectively, were ultimately merged. The effort at unification began in 1994 and resulted in the creation of the Unified Modeling Language (UML) in 1997. The language is periodically updated and revised by the design team and a consortium that was formed for this purpose (Booch, 1999, pp.xviii-xx).

Objects in the ArcGIS Hydro data model are designed as components of a UML static structure diagram. These elements are called classes, and are represented as three-component boxes in the UML diagram. The UML allows for classes to inherit properties from one another and to be in relationships with each other. Child classes who participate in an inheritance relationship with a parent are called subclasses. Classes that take advantage of non-inheritance relationships to add detail to a class are called subtypes. The ArcGIS Hydro data model makes use of both subclasses and subtypes in the UML diagram to streamline the schema

and to conform to good modeling guidelines. Subclasses are children of a parent class, and each child inherits all the properties and behaviors of the parent, but may also have any unique properties of its own which are required to describe it. Subclasses are used to describe features which have different behavior. Subtypes are different flavors of a parent class that all have the same behavior. Subtypes are discrete instances of the parent class, not children of it. They may have different domains for each of their attributes, but all subtypes of a given class have the same attributes. Subtypes are also involved in connectivity rules.

3.2.4 Computer Aided Software Engineering (CASE) Tools

Computer Aided Software Engineering (CASE) tools are programs used to facilitate software design, much as a Computer Aided Design (CAD) tool is used to facilitate structural design. The CASE tool exports UML diagrams to an industry standard format called Microsoft Repository. The Repository is read by ArcCatalog to create the schema so data may be loaded into it or to apply the schema to existing data.

The use of CASE tools to develop a data model has made a few things possible. First, it makes sharing the model easy. If the schema were created directly in ArcInfo, it would be difficult to share with and transmit to others. Second, it makes the model easier to explain. The diagram form of the model is much easier to comprehend than the text description. A text description certainly answers questions regarding specifics of the attributes and object definitions, but the structure of the model is more quickly grasped from the diagram. Third, it enables automatic code generation, which allows users to create objects and add

behavior through C++ code. The Repository is a Microsoft Access database that contains all of the information from the UML diagram. It makes the information from the UML diagram accessible to ArcCatalog as well as the code generation tool.

3.2.5 Component Object Modeling (COM)

One of the benefits provided by CASE tools is that they allow software diagrams to be automatically converted to program code. This code is compiled to become a .dll file, which is what the computer uses to create objects. Objects can be coded using COM, or Component Object Modeling, which is a programming standard promoted by Microsoft to foster cooperative development for the Windows environment and to provide a framework for programs to interact with one another and share data. COM-compliant languages (like Microsoft Visual C++ or Visual Basic) are used to build programs and objects that conform to the standard. Because of COM, suites of office software can be built so that all the components interface in a standard way and can acquire objects from each other, and GIS programs can share data objects with modeling software in the same way. COM-compliant geobjects are built so that code may be applied to them and manipulate them, whether in the context of ArcGIS or some other program. This frees users from the necessity of learning proprietary languages like Avenue and AML, which were formerly required to customize ESRI products, and allows easier development of software which will build on the functionality of ArcGIS.

3.3 NETWORKS

3.3.1 Components

Lines forming a network are called Edges, and their intersections are called Junctions. Both Edges and Junctions come in two variants, simple and complex. A Simple Edge is a line segment that connects exactly two Junctions. A Complex Edge is a line that may connect more than two Junctions. Simple Junctions are points at which the network may be closed off, as with a valve. Complex Junctions are collections of Junctions and Edges that act as one entity, such as a switch box on an electrical network. Many Edges may meet at a Junction, but at least one is required.

Edges and Junctions are similar to Arcs and Nodes. However, the topological relationships characteristic of ARC/INFO 7 do not exist in ArcGIS. The network feature classes called Edges and Junctions in ArcGIS have a relationship system that is similar to topology but is more lightweight. Topology serves to establish connectivity, contiguity, containment relationships among features (AGI, 1996, “topologically structured data”). Contiguity means that features are adjacent to each other, and containment means that one feature exists completely within another. The only part of topology that is carried into the ArcGIS network system is the connectivity relationship.

As shown in Figure 3.4, Simple and Complex Edges are derived from a general feature called an Edge, which is in turn derived from an even more general Network Feature. Network Feature is also the parent class of Junction, which in turn is the parent class of Simple and Complex Junctions. Network

Feature, Junction Feature, and Edge Feature are all titled in italics in Figure 3.4, while Row, Object, Feature, and all the other classes shown are in plain type. Italics indicate an Abstract Class in UML, a class which cannot be instantiated. Abstract classes exist as anchors for generalization in the model and divide the model into logical sections. Properties and methods are carried higher up in the diagram tree by abstract classes so they do not have to be repeated on numerous child classes.

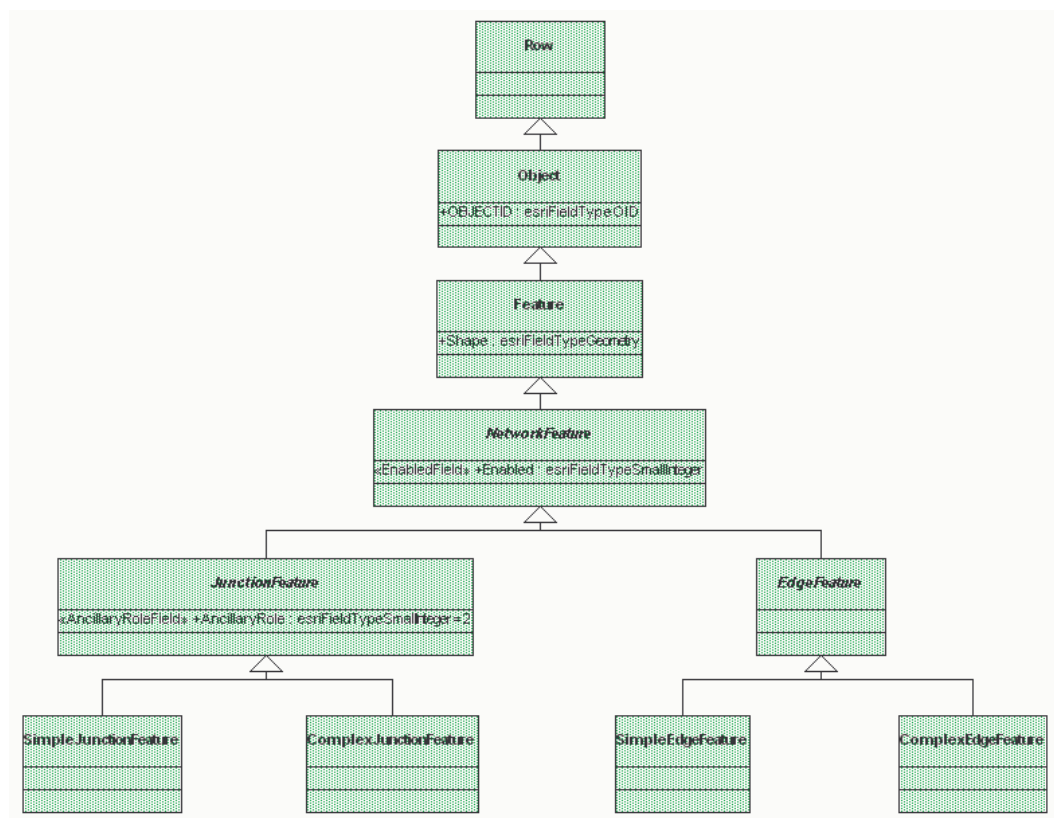


Figure 3.4 The ArcObject set showing the derivation of Simple and Complex Edge and Junction objects

A common example of Simple and Complex Edges is in water supply. A trunk line carrying water through a neighborhood, as in Figure 3.5 would commonly be represented as a Complex Edge, and at many Junctions along its length, small tap lines would run from it to houses. The small lines usually run straight from the trunk line to a building and there is no cause for them to have interior Junctions. However, if the trunk line were not a Complex Edge, it would be broken into possibly hundreds of separate Simple Edges at its Junctions with supply lines, each piece requiring its own a row in a database table, complicating maintenance of attribute data on the large line.

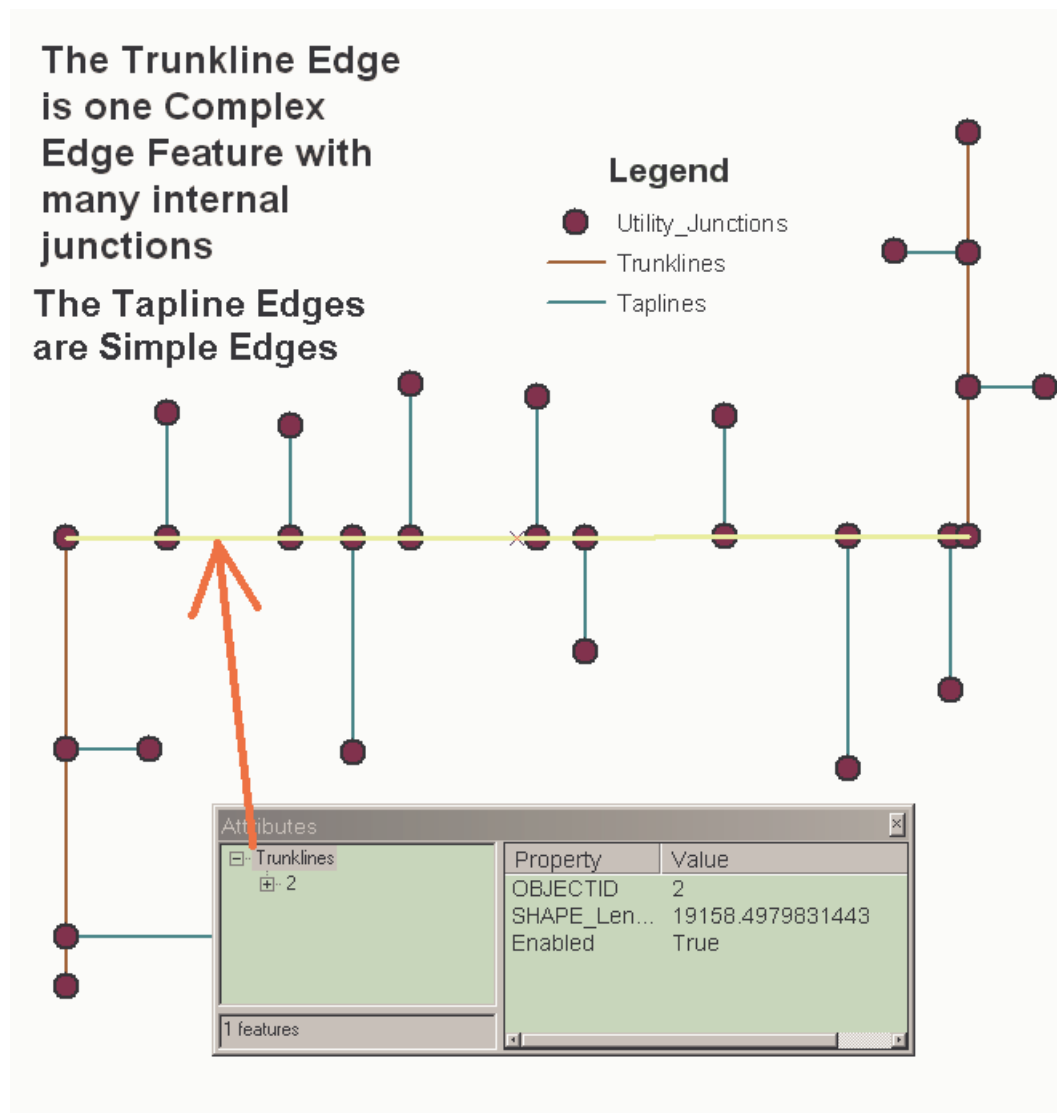


Figure 3.5 Example of Simple and Complex Edges

Junctions also come in simple and complex varieties. Simple Junctions are used exclusively throughout the ArcGIS Hydro data model. They are comparable to the nodes of ARC/INFO and are represented as points. Complex Junctions are intended for use in situations like electrical networks, where at one

point on the map complex algorithms determine the direction of flow. These cannot be created in simple graphical fashion and require some C++ code to instantiate. They might be important to some users (for example in representing the gates on a dam), so it is left to the analyst customizing the data model to decide if Complex Junctions are necessary in their implementation.

3.3.2 Conceptual Framework for Networks

Modeling relationships is one of the new features of ArcInfo 8, and the most useful relationships for this work have been those for network connectivity. A network is not a single entity. Rather, it is a composite of three models, one describing the network features, one describing the relationships among them, and the third describing how the network relates to other features. The three constituent models are the logical model, the geometric model, and the addressing model.

Network connectivity is established and maintained by a table referred to as the *logical network*. This jagged-edge table, shown in Figure 3.6, contains a list of Junctions and their connections to other Junctions through Edges. Essentially, the logical model describes what is connected to what and in what sequence. Any number of Point, Line, Edge, and Junction feature classes can be incorporated into a single logical network, but each feature class may be a member of only one logical network. As long as they are members of a network, Point and Line classes exhibit the behavior of Edges and Junctions.

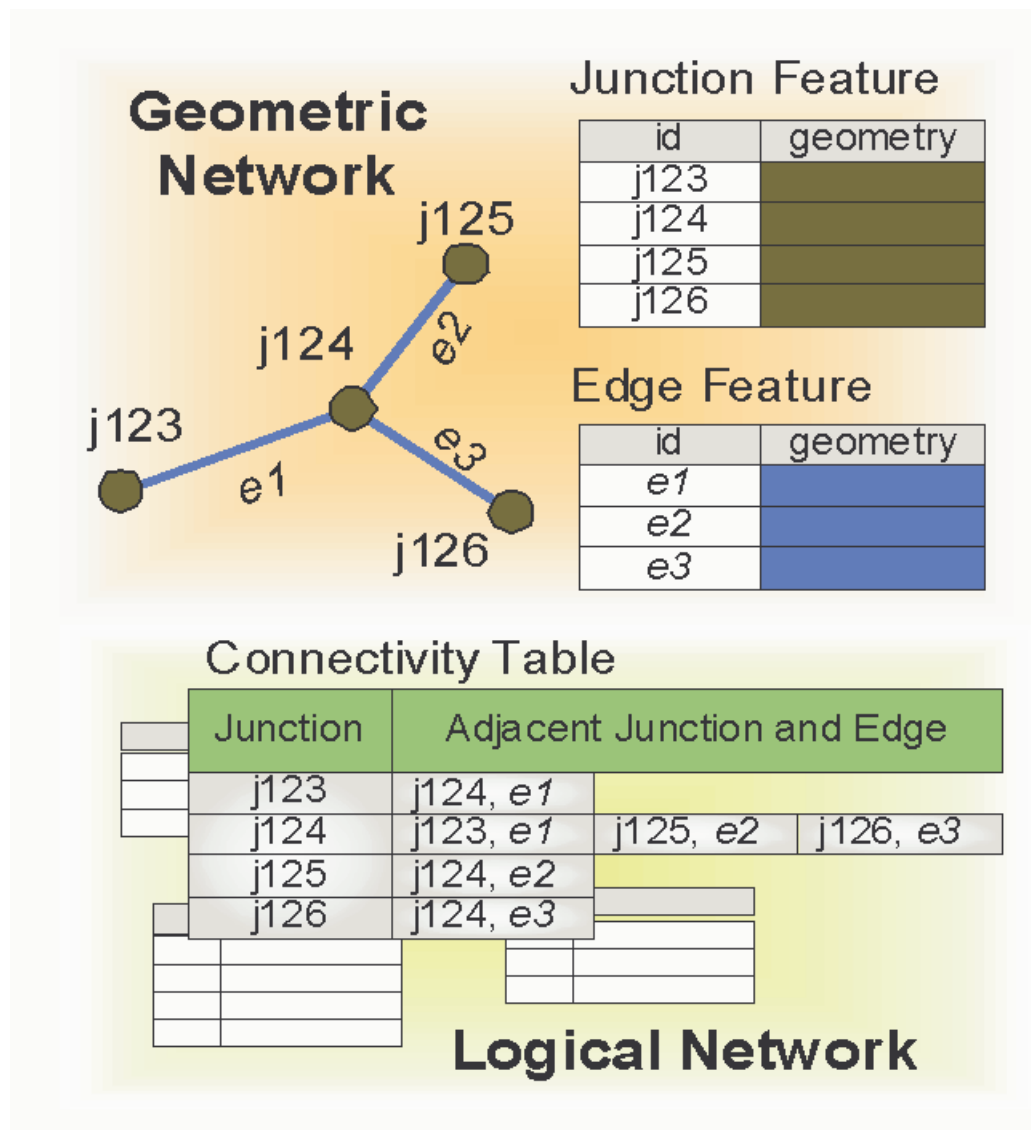


Figure 3.6 Diagram of the geometric network and logical network tables (Zeiler, 1999, p.132)

The *geometric model* describes where the network features are actually located in space, i.e., what their x and y coordinates are. It is a collection of the tables of component feature classes that participate in the logical network, as

shown in Figure 3.6. When a dataset is projected from one coordinate system to another, its geometric model changes, but its logical model of feature connectivity is unaltered. Whenever features in the network are edited significantly (so as to change their connectivity with neighboring elements), the logical network must be reconstructed from the new geometric network. For this reason, any time the network's geometric components are edited, the logical network must be open in ArcMap and available for editing.

The *addressing model* defines how locations fall on the network. Suppose a gage station has a particular latitude and longitude. This locates the gage adjacent to a specific river reach and at a certain distance along the reach. The “river address” defined by pairing the reach and a distance along it enables tracing upstream and downstream along the network to identify other point and line features on the river network whose functioning may affect the measurements at the gage station.

3.3.3 Linear Referencing

Specifying locations along a river or stream as an address on the network, rather than a pair of Cartesian coordinates is analogous to specifying that a house is located at 123 Oak Ave, rather than giving its latitude and longitude. The network components of the ArcGIS Hydro data model, collectively known as the Hydro Network, can be marked off with numbers using the addressing model. Just as a street network is marked off and transportation routes planned through it, the Hydro Network can be navigated according to network addresses.

In the National Hydrography Dataset, a two-dimensional addressing scheme is used. Point locations on a reach are described by a Rch_Code (for Reach Code) and a percentage of the distance from the downstream end of the reach. The Rch_Code is a 14 digit number made up of an 8 digit Cataloging Unit number and a 6 digit segment number unique within the Cataloging Unit which are concatenated. With such an addressing system, it is possible to precisely define the location of a point on the network without reference to any geographic coordinate system, and the point is guaranteed to coincide exactly with a reach.

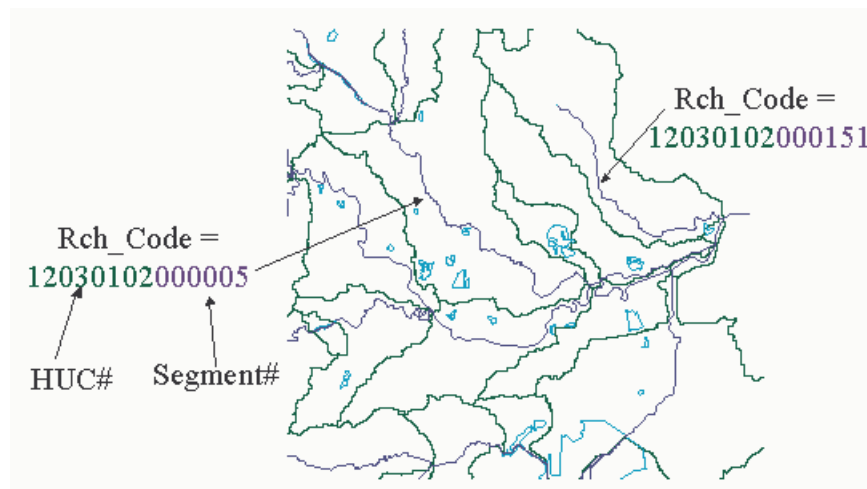


Figure 3.7 Construction of the Rch_Code for linear referencing

The LLID (Longitude/Latitude Identifier) system used to identify rivers in the Oregon and Washington Hydrography Framework employs a two-dimensional scheme also, referring to point locations by citing the LLID of the stream on which the point is located and the distance from the mouth of the stream in map units. The LLID is a 13-character string composed of “the concatenated decimal degree longitude and latitude of the feature” (Washington State Department of

Ecology, 2000, p.8). The coordinates are in degrees, minutes, and seconds, with the first seven digits for the longitude of the river mouth, and the next six for latitude, as shown in Figure 3.8. This system also uses upstream measures from the mouth of a river, but instead of using a percentile distance, it uses the absolute distance.

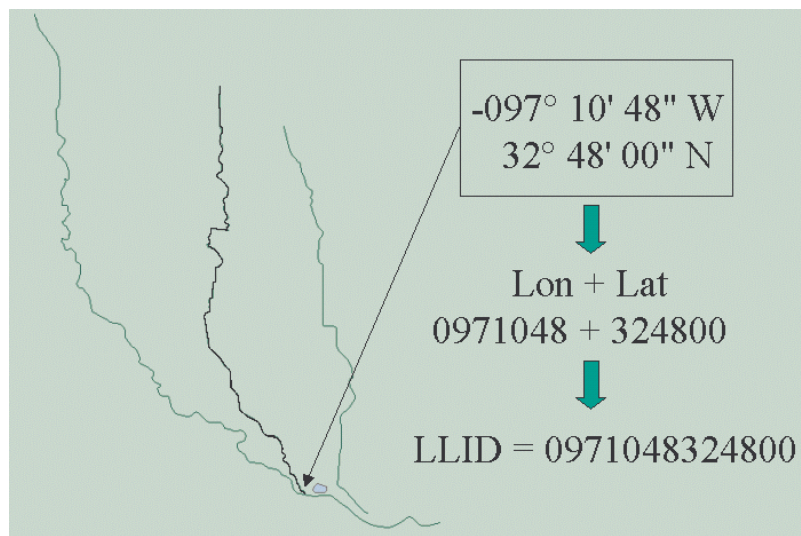


Figure 3.8 Construction of the LLID for linear referencing

The addressing system of the ArcGIS Hydro data model is two-dimensional, because it uses unique numbers for each Hydro Edge, like the LLID system, and does not need to specify which Watershed the reach falls in. The systems are compatible because at the heart all three refer to a particular Edge and a distance along it, although the NHD uses two coordinates to designate the Edge. Converting among the three systems is not trivial, however, so it is suggested that users make a careful study of maintenance and usage needs before adopting a linear referencing system.

The network addresses, called *Measures*, m-values, or m's, are stored on the Edges along with the coordinates of the vertices as a coordinate triplet (x, y, m) for each vertex. Measures can be assigned in one of two ways. The numbers can run from 0 to 100 and indicate the percent of total line length at which they are located, or they can run from M to N where (M - N) is the length of the line and represents distance along it in map units. The percent system is commonly used and may be referred to as a Relative Addressing system. The actual length system is referred to as absolute distance, or Absolute Addressing. Both systems have advantages and disadvantages related to scale of the map and the effects of editing. Both systems are usable with the ArcGIS Hydro data model.



Figure 3.9 An illustration of the differences between Relative and Absolute Addressing

3.4 ARCGIS HYDRO DATA MODEL

The UML diagram of the ArcGIS Hydro data model is divided into four logical sections. The model is made of Hydro Network Features, Hydro Features, Channel Features, and Time Series. The Hydro Network is the heart of the model; all the other pieces of the model depend on it in some way, or exist to support it. Hydro Feature classes serve two purposes. They act as a temporary storage location for Features to be converted to Network Features, and they hold auxiliary data which improves the quality of analysis tasks. Channel Features are those required to describe a river channel in three dimensions. They support flood inundation modeling and storm water routing studies. Time Series data are the collections of observations along the river network. These features are still under development and their structure is not finalized.

The following sections are an object dictionary for the ArcGIS Hydro data model. In each subsection dealing with a major feature class, the class will be briefly introduced, all its attributes will be defined, the class will be discussed in depth, and finally the class will be illustrated with a UML static structure diagram.

3.4.1 Hydro Network

The Hydro Network is composed of Hydro Edges, Hydro Junctions, Waterbodies, and Catchments. It is the heart of the ArcGIS Hydro data model and is built on the functionality of ArcInfo's Utility Network Analysis. This involves certain rules and restrictions for the features involved, and allows certain analysis tasks which are made possible by those rules.

3.4.1.1 *Hydro Edges*

Hydro Edge is the parent class representing most of the water features in the object model. These features are typically represented in hydrography as a blue line. Hydro Edge inherits from Simple Edge Feature (an ESRI Network Feature) and carries flow through the network. Figure 3.11 shows the UML static structure diagram for Hydro Edge.

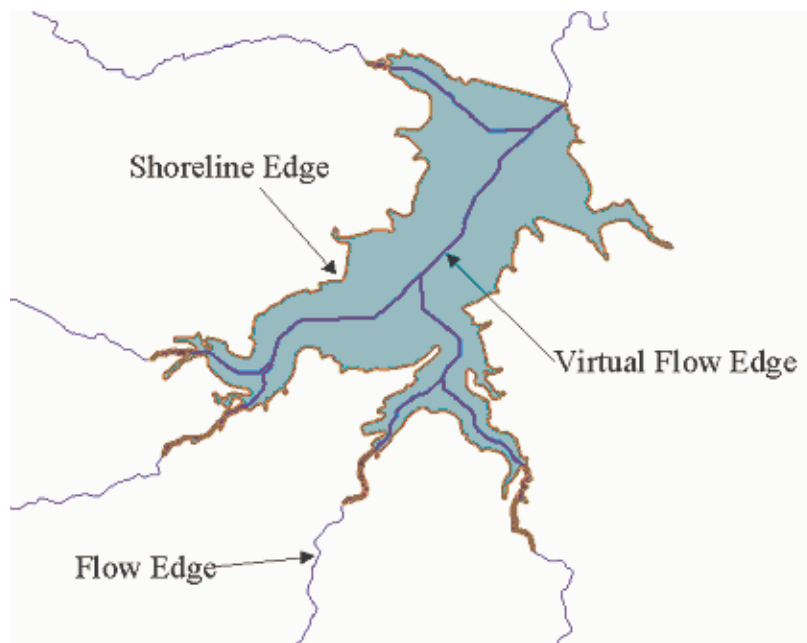


Figure 3.10 Examples of the three subtypes of Hydro Edge

Flow Edges are the primary components of the network. They represent such features as streams, rivers, canals, ditches, and pipelines that can be represented by a single line on a map. *Virtual Flow Edges* represent the path of water flow through water bodies such as lakes, swamps, bays, estuaries, and wide rivers normally shown as shaded blue areas on maps. The Virtual Flow Edge is

the flow centerline or thalweg of the water body in question, the line representing the path of water from the inlets of the water body to the outlet(s). Virtual Flow Edges are used to bridge the gaps in the network created by on-channel waterbodies. Flow tracing algorithms cannot trace through a polygon because it does not participate in the logical network. Therefore, Virtual Flow Edges replace the waterbody and help create a continuous flow network. *Shoreline Edges* are the interfaces between water and land. They represent the shorelines of water bodies, the banks of wide rivers, and the perimeters of islands and coastlines. Shorelines do not carry flow in the network, flows assigned to them are passed on to the first Flow Edge or Virtual Flow Edge encountered on a shoreline. Shoreline Edges are used to derive the Edge Catchment for the water body.

Hydro Edge has three subtypes, as shown in Figure 3.10. The subtypes of Hydro Edge are Flow Edge, Virtual Flow Edge, and Shoreline Edge. Hydro Edge has three attributes: LinearRef_ID, LengthInMeters, and HydroEdgeType. They are defined as:

LinearRef_ID-A user-defined identification number for the feature which can be used to support linear referencing. This need not be unique to the Hydro Edge. An ideal use for this field is to store agency-assigned ID numbers, such as Rch_Code from the USGS NHD dataset. This ID number is used as a key value in relationships to tie together the Shoreline Edges, Virtual Flow Edges, Waterbody polygons, and various Hydro Junctions that all represent a water body entity.

LengthInMeters-The length of the feature calculated in meters in an Albers equal area projection. This is created so that the dataset can be stored and used in geographic coordinates without loss of true lengths for the network Edges.

HydroEdgeType-An integer code designating the Hydro Edge as either a Flow Edge (1), Virtual Flow Edge (2), or Shoreline Edge (3). The presence of subtypes to classify the Hydro Edge class allows for the creation of connectivity rules for each subtype. Connectivity rules define what kinds of Junctions connect particular types of Hydro Edges

Hydro Edges can be related to each other through the LinearRef_ID to link them into longer segments. For example, the River definition used in the Washington Hydrography Framework starts at the headwater and keeps the same identifier, called the LLID, the entire length of the river, which traverses many Hydro Edges. Using LinearRef_ID to link Edges and to create events on them gets around the limitation of Simple Edges which causes each Hydro Edge to be defined from confluence to confluence.

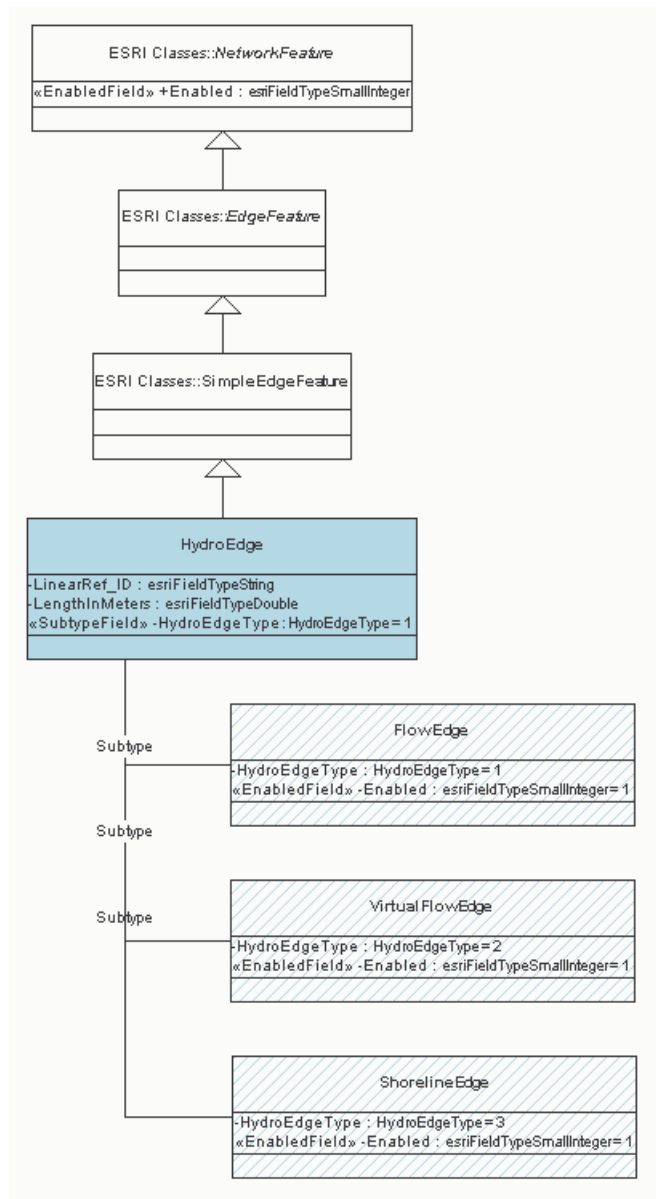


Figure 3.11 UML diagram for Hydro Edge

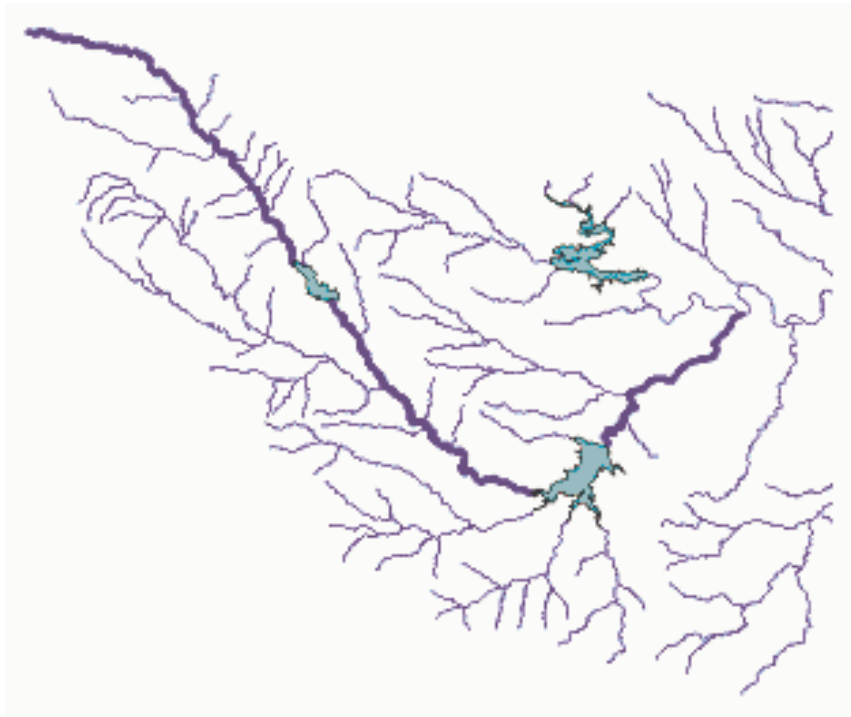


Figure 3.12 Use of the LinearRef_ID to link Flow Edges

Use of simple network features

All Edges in the Arc Hydro Data Model are Simple Edges. A Simple Edge is a line segment that begins at one Junction and ends at another. It may not have branches and it must be contiguous. A Complex Edge is a line segment that begins and ends at Junctions, but it may have internal Junctions along its length where it may intersect other Edges. The decision to use only Simple Edges in the model was made for two reasons. First, Simple Edges better support linear referencing. If Edges were allowed to be complex, the rule system for assigning Measures would be complicated, with many exceptions. The process of maintaining route measures after editing the Edges would likewise be difficult.

Second, Edge Catchments are in a one-to-one relationship with Hydro Edges. This allows network solvers to handle the properties of the land surface and associate them with the Hydro Network for analysis. If the Edges were complex, the result of a catchment delineation operation would no longer be a one-to-one relation between Hydro Edges and Edge Catchments. Rainfall routing would not work on such a system. For these reasons, the Hydro Edges are derived from Simple Edge features, rather than Complex Edge features.

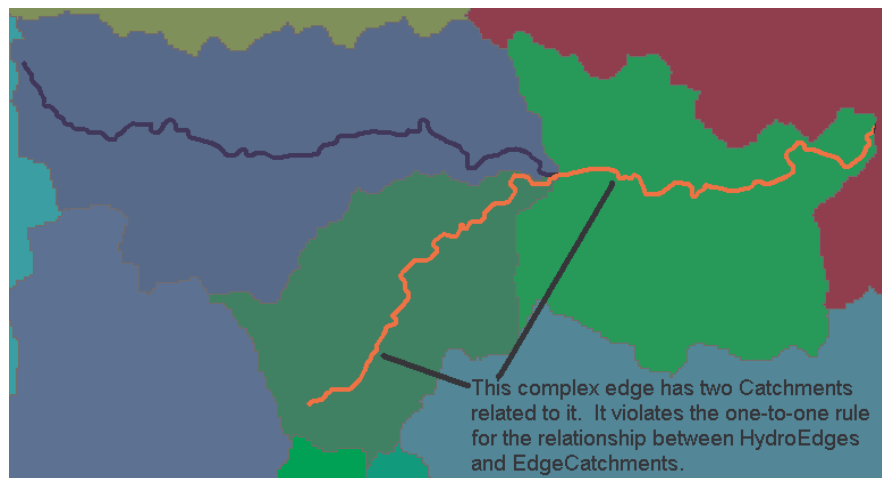


Figure 3.13 An example of why Hydro Edge derives from Simple Edge

3.4.1.2 *Waterbodies*

A *Waterbody* is defined to be a region of contiguous water represented as an area on a map. This definition includes typical features like lakes, swamps, estuaries and marshes, bays, oceans and ponds. It may also (at the user's discretion) include wide rivers that are depicted on maps by a pair of bank lines. Shoreline Edges can be derived from the perimeter of the polygon that represents

the water body. Waterbody has the attribute Wbody_ID, which supports the relationship between Hydro Edge and Waterbody.

Waterbody, as shown in Figure 3.14, has a Wbody_ID attribute and Hydro Edge has a LinearRef_ID attribute that are intended to store the identifiers assigned by users or by agencies, like the NHD Rch_Code or the LLID used in the Washington Hydrography Framework. These can be used to group Waterbodies and Hydro Edges together in a tabular fashion. In such a situation, all the Hydro Edges (Flow Edge, Virtual Edge, and Shoreline Edge) that traverse or bound a water body may have the same LinearRef_ID, which matches the Wbody_ID.

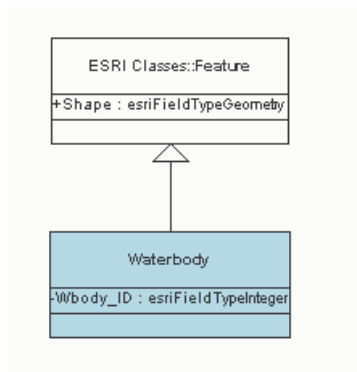


Figure 3.14 The UML Diagram for Waterbody

Waterbody also relates to a type of Hydro Feature called a Closure Line. This is a line that closes off one water body from another, in locations where they are contiguous, as in Figure 3.15. This type of situation arises in large lakes or bay systems that are not separated by land, but for various reasons, either cartographic, regulatory, or by local custom, are treated as separate water bodies.

The closure line in conjunction with Shoreline Edge forms a closed figure around the water body.

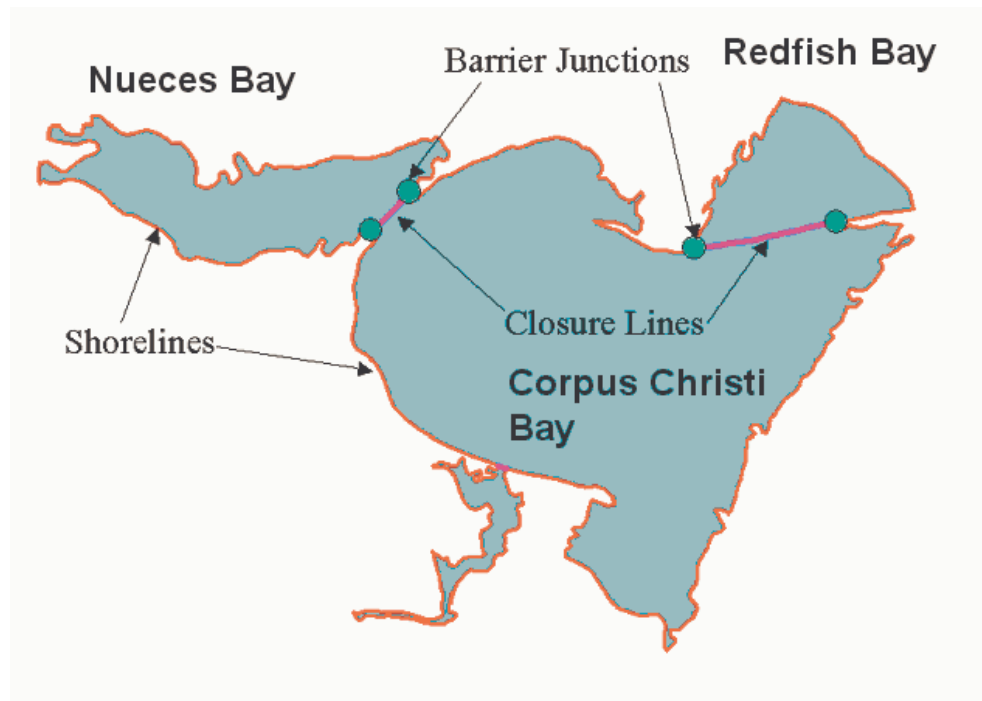


Figure 3.15 An example of Closure Lines separating Waterbodies

3.4.1.3 *Hydro Junctions*

Junctions are the locations at which Hydro Edges intersect one another. Most of the time, such locations are anonymous points on the network, with no attributes or user interests. These are stored in a feature class called ESRI Generic Junction. The presence of these Junctions is required for network connectivity. They are very similar to the nodes in ARC/INFO 7.

Junctions other than the generic ones may have particular behaviors and attributes, some assigned by the user, and others required by the ArcGIS Hydro

data model. The special Junctions in the ArcGIS Hydro data model are the Virtual Junction, Shoreline Junction, Sink, and Barrier Junction. These four are subtypes of the class Hydro Junction. ArcGIS database validation procedures require that connectivity rules be established between subtypes in the network. Connectivity rules are explained further in section 3.4.5.

Virtual Junction is the subtype created to mark the intersections of Virtual Flow Edges found in complex Waterbodies. Virtual Junctions are used to enforce the connectivity rules between Virtual Flow Edges and to ensure database integrity.

A Shoreline Junction exists between a Virtual Flow Edge and a Flow Edge, denoting the entrance of the Flow Edge into a Waterbody. They also serve as crossroads, forming the coincident endpoints of two Shoreline Edges. On the boundary of an island polygon, this Junction is not used; the network construction algorithm will create an ESRI Generic Junction at such a point.

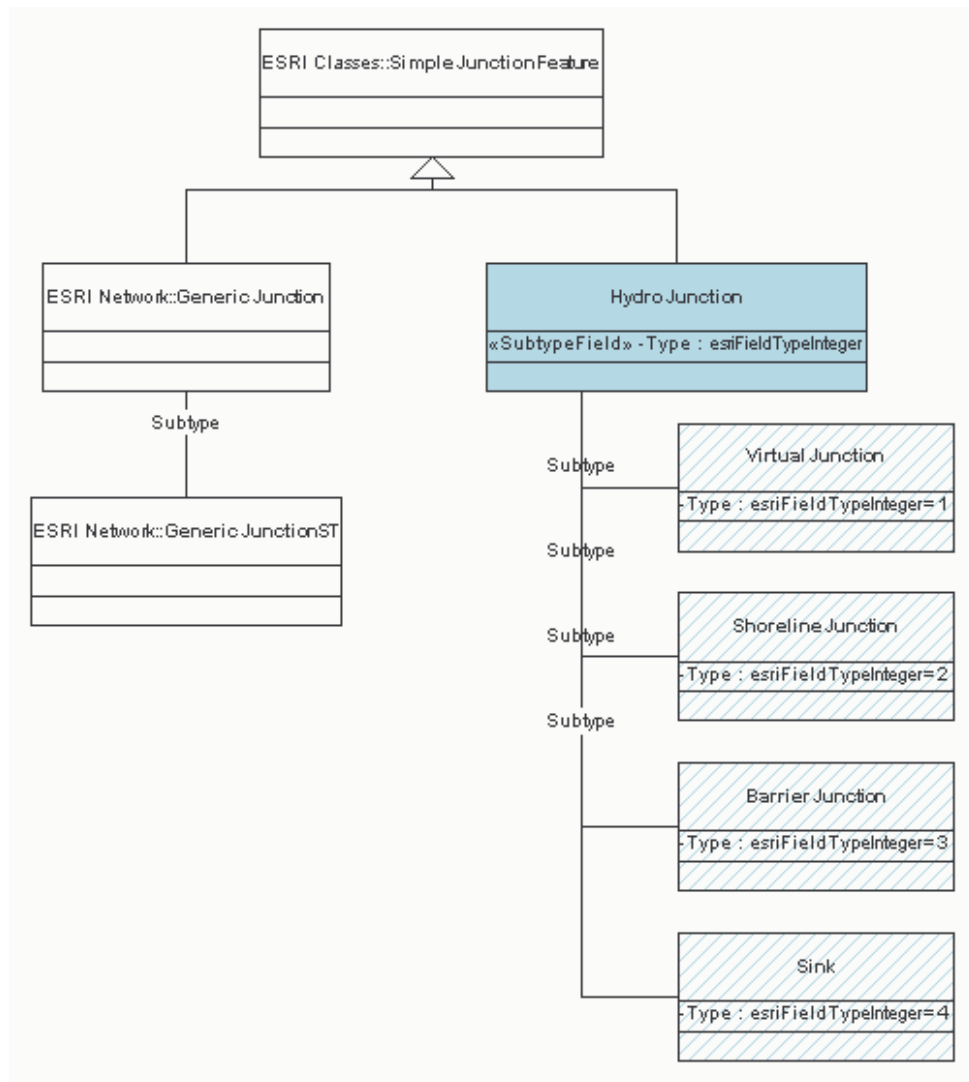


Figure 3.16 The UML Diagram for Hydro Junction

Barrier Junctions are a subtype of Hydro Junction used on Shoreline Edges to mark the boundary between contiguous Waterbodies. These Junctions serve as a barrier for solvers that perform flow traces so the flow will not

inadvertently be traced from one water body to another, or along riverbanks, for great distances without entering the flow network.

Sinks are the outlets of networks or sub-networks, created to work with the flow direction assignment solver. Flow on network edges is always directed toward a sink, regardless of the intervening distance. As a result, a single sink may be set at the outlet of a river basin and every stream in the basin will have a downstream flow direction because all the flow will be directed to the basin outlet.

ESRI Generic Junction is a feature class that exists for the network creation solver to store the “orphan junctions” it automatically creates. These Junctions are created at every Edge endpoint or intersection where a user-defined Junction does not exist. ESRI Generic Junction ST is a subtype of the network-created orphan Junctions. The subtype is allowed to participate in connectivity rules within the network so that database integrity can be enforced.

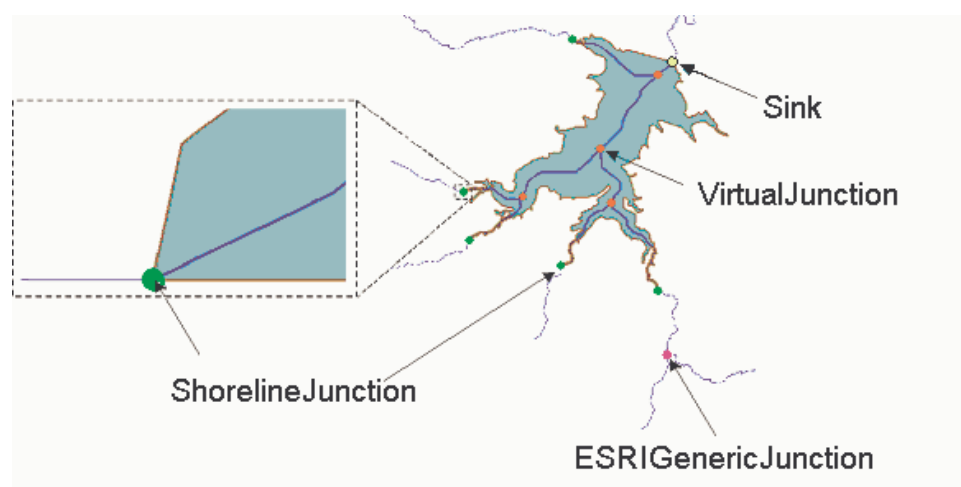


Figure 3.17 An example of the three main subtypes of Hydro Junction and ESRI Generic Junction

3.4.1.4 Hydro Events

Hydro Events describe information located on the Hydro Network by linearly referenced addresses. Hydro Event has one attribute, the LinearRef_ID. It is assumed the user will add other attributes to this, such as an agency-assigned identifier field, or a comment field.

Hydro Event has two child classes representing point and line events. Hydro Point Events are located on a Hydro Edge by the LinearRef_ID, and at a particular point on the Edge by the value stored in the Measure attribute field. Hydro Point Events are useful for locating features that will always be located exactly on a Hydro Edge.

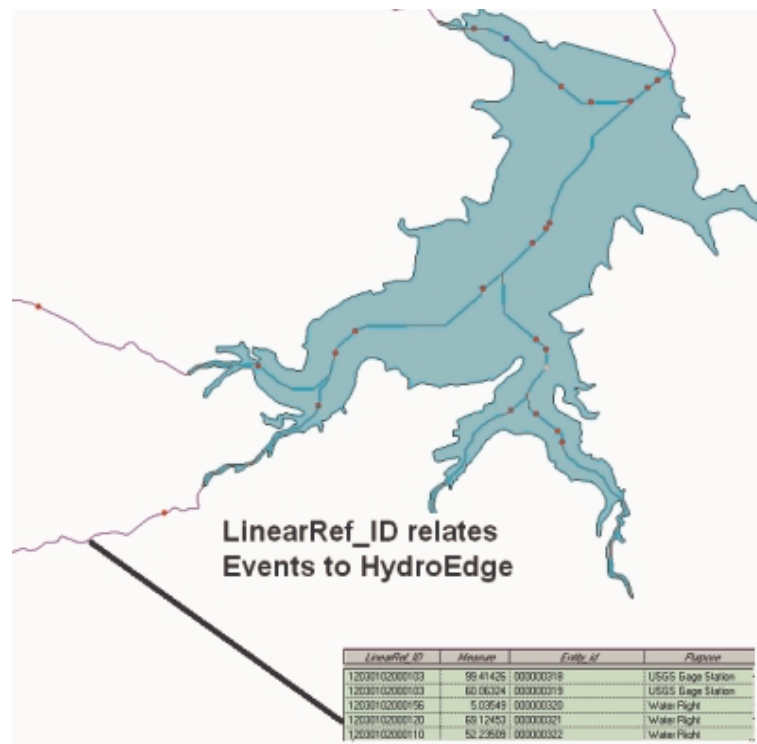


Figure 3.18 Examples of Hydro Point Events on Benbrook Lake in the Lower West Fork of the Trinity River

Hydro Line Events are located similarly, on a Hydro Edge by the LinearRef_ID inherited from Hydro Event, but they run along the Hydro Edge for a particular length. They are defined by their attributes: LinearRef_ID, FromMeasure, and ToMeasure.

LinearRef_ID-A number uniting the various elements of an event that spans more than one Hydro Edge. Note that this allows both noncontiguous and branching line events to be constructed.

FromMeasure-The M Value along the Hydro Edge at which the Hydro Line Event begins

ToMeasure-The M Value along the Hydro Edge at which the Hydro Line Event ends

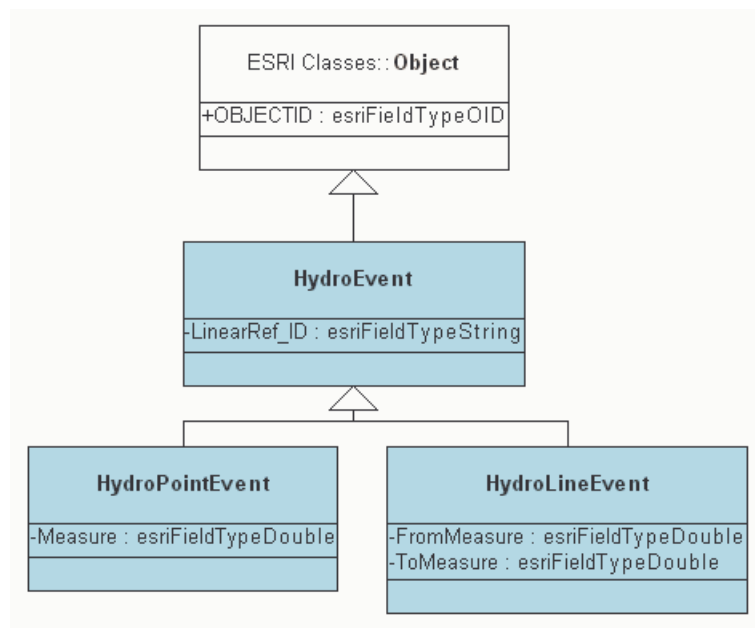


Figure 3.19 The UML diagram for HydroEvents

Hydro Events are derived from Object, not from Feature as most other classes in the ArcGIS Hydro data model are. The difference is that Features are Objects with a Shape attribute. Events do not have a shape stored in the table; the LinearRef_ID and Measure(s) in the table describe how to draw the events *given the Hydro Edge class* as a foundation. An Event table requires less disk space than one that stores features. Further, if the Hydro Edge is reprojected into a different coordinate system, the event table still matches it without need of reprojection. However, Hydro Events must be coincident with an Edge. Furthermore, unless they are converted into Feature Classes, Hydro Events are not displayable without the Edge class from which their addressing model is built.

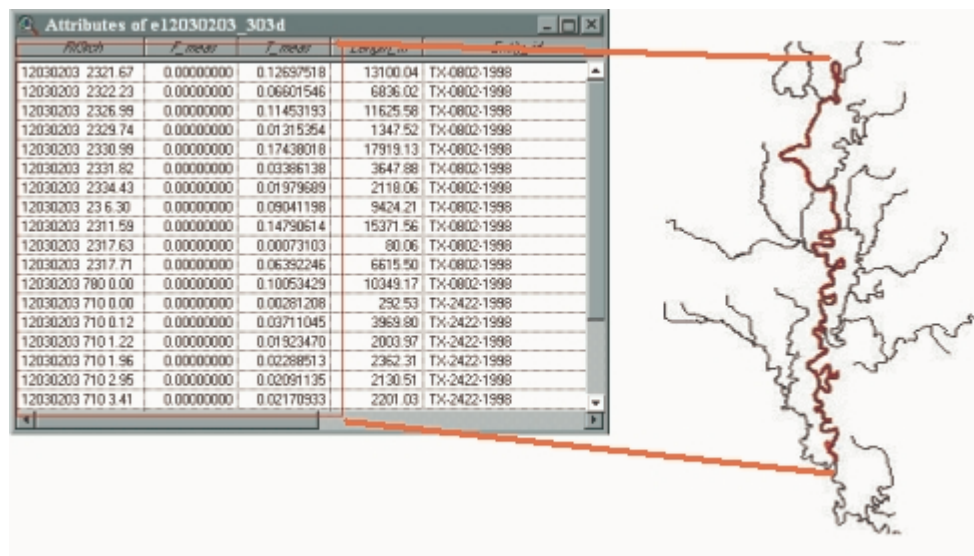


Figure 3.20 An example of a linear event near the mouth of the Trinity River represented in the table by the boxed fields and in the map by a red line.

3.4.1.5 *Catchments and Watersheds*

Edge Catchments are portions of the landscape that drain to a particular Edge. There is a one-to-one relationship between Edge Catchments and Flow Edges, so that every catchment has one Edge, and vice versa. Watersheds are a collection of aggregated Catchments. They are used to store Hydrologic Units that are part of the NHD, and any large drainage areas users need.

Edge Catchments have the properties GridCode, Wshed_ID, AreaInSqMeters, and Wbody_ID. GridCode is a relic of the catchment creation process. The polygons representing catchments are created by passing the Watershed request to an appropriately prepared digital elevation model grid. This request returns a grid of watersheds that each have a unique value corresponding to the Edge that drains them. GridCode is simply an ID field that links the Edge Catchment to the Edge from which it is derived. Wshed_ID is a field that can be used to aggregate Edge Catchments into non-overlapping Watersheds. A group of adjacent Edge Catchments are aggregated into a single Watershed, and Wshed_ID links the component Catchments to the Watershed. AreaInSqMeters is the area of the Edge Catchment polygon in square meters. This attribute is maintained so that if the network is in geographic coordinates or is distorted by projection into a new coordinate system, the correct area of the Edge Catchments will be available. The Wbody_ID field is used to link Edge Catchments that drain to Shoreline Edges to the appropriate Waterbody.

Watersheds have only the attribute Wshed_ID. This attribute can be used to create nested, non-overlapping Watersheds by aggregating smaller Watersheds

into larger ones. Wshed_ID always points to the next Watershed up the nesting hierarchy. This type of nesting system is the basis of the Hydrologic Units of the NHD. In the largest Watersheds, at the highest level of nesting, the Wshed_ID is either set equal to the ObjectID of the Watershed, or it is set to some null value to indicate that there is no higher level Watershed.

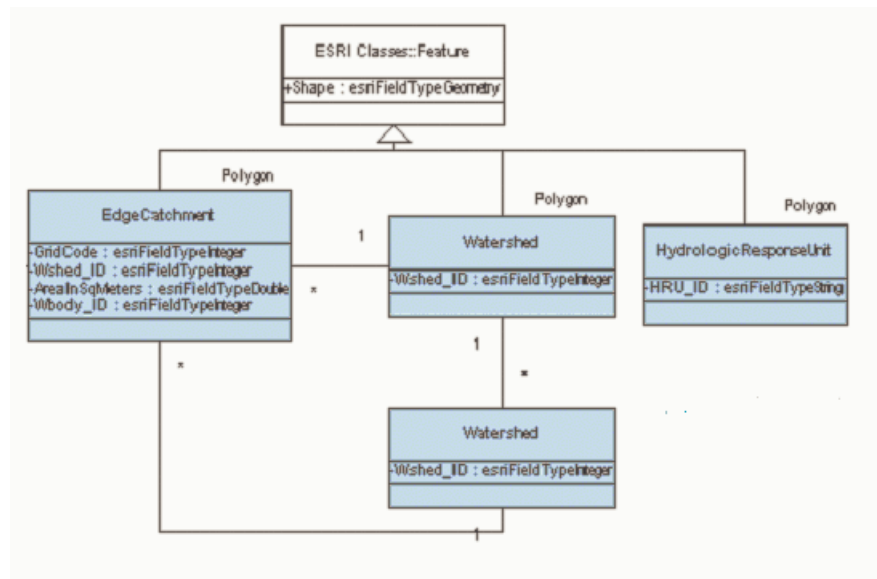


Figure 3.21 UML diagram of Watershed, Catchment and Hydrologic Response Unit

3.4.1.6 Hydrologic Response Units

Hydrologic Response Units are polygon features related to Watersheds and Catchments. The purpose of tying Edge Catchments to Flow Edges is to make properties of the land surface, which are significant to hydrologic modeling, accessible to the Flow Edges and any programs that access them. The properties of the landscape are stored either in grid format or as polygon files. These data

about the landscape are stored as Hydrologic Response Units. The landscape data of Hydrologic Response Units is typically related to the vertical transfer of water in the hydrologic cycle. These data will pertain to the passage of water from the atmosphere to the land, from the land to the subsurface, or from one part of the land surface to another. A typical example of hydrologic response units is shown in Figure 3.22, which depicts the landuse zones of the city of Austin, TX. Certain landuses, e.g. commercial lots, contribute more pollution and more runoff during storm events, while others, e.g. water, contribute none. Using these Hydrologic Response Units, planners can predict the expected levels of pollution and amounts of runoff, or test the potential effect of cleanup measures in hypothetical storms. The Hydrologic Response Units are intersected with Edge Catchments to determine values of hydrologic response data within a particular Edge Catchment. An average value can then be stored with the Edge Catchment or the Flow Edge, or both.

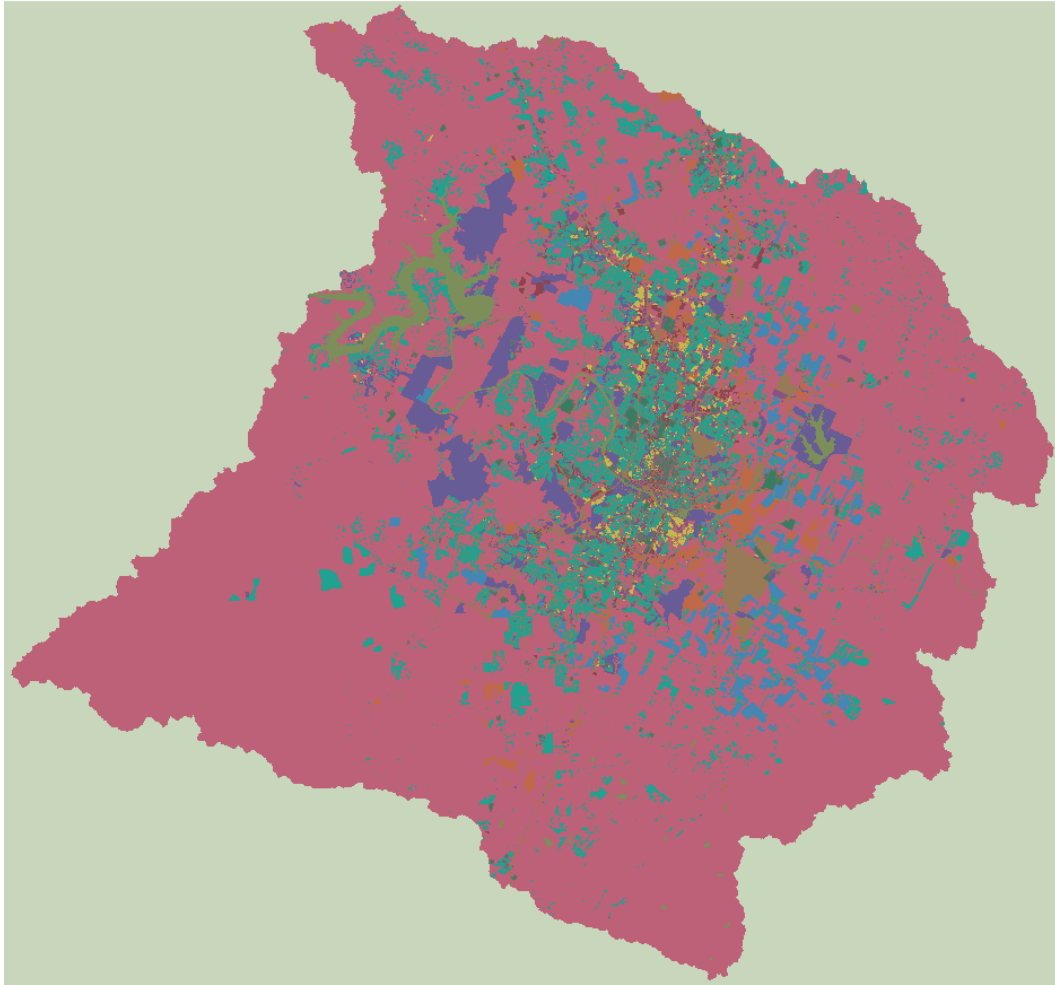


Figure 3.22 Hydrologic Response Units across the city of Austin representing landuse zones

3.4.2 Hydro Features

Hydro Features are the classes that hold the descriptive cartography that supplements the other portions of the data model. Any points, lines, or areas that are not part of the Hydro Network or the Catchments are stored as Hydro Features. These include such data layers as governmental boundary lines,

landmarks, off-network water bodies, and cultural features. Hydro Features also hold the simple line and point features which are converted to network Edges and Junctions as the data model is constructed. Because there are so many different kinds of Hydro Features, the data model does not attempt an exhaustive description of them.

Feature is the parent class for Hydro Features. A feature is any row in a table that has a unique identifier and a shape. Shapes can be 0-dimensional (points), 1-dimensional (lines) or 2-dimensional (areas). In the ArcGIS Hydro data model, there is a feature class for each dimension: *Hydro Points*, *Hydro Lines*, and *Hydro Areas*. The Hydro Point class has four child classes, Monitoring Point, Flow Change Point, Structure, and User Point. The User Point class is meant to hold whatever types of point features do not fit into the other classes.

The National Hydrography Dataset (NHD) includes many features that map directly to the classes of Hydro Features. Special Use Zones, Special Use Zone Limits, Snag/Stumps, Dam/Weirs, Fumaroles, Spring/Seeps, Fish Ladders, Anchorages, and more all fall under Hydro Features. The general rule is that all NHD features which are not reaches become Hydro Features in the ArcGIS Hydro data model.

Some NHD features have more than one possible spatial representation (e.g.: dams can be areas or lines, depending on map scale) so they may need to be represented in more than one Hydro Features class. This is entirely acceptable and expected. If the user desires a relationship between such features, it is left to

the user to create the relationship. Ordinarily, a simple table join will suffice to create the desired link between the two feature representations of the object.



Figure 3.23 Hydro Features from the National Hydrography Dataset (NHD)

Some users may find that they do not need access to regional or national datasets. They may have a particular need for locally generated data. A typical example of this is a city that used aerial photography to gather data and then digitized its own stream network. The original line work from those types of data goes into the Hydro Features class Hydro Line, and then the centerlines of streams are converted to network types Flow Edges, Virtual Flow Edges, etc. The other data digitized, such as bank lines or floodplain extent, remain in the Hydro Line class, however.



Figure 3.24 Aerial photography and derived stream network

3.4.2.1 *Hydro Points*

There are four child classes, or subclasses, of Hydro Points, Structures, Flow Change Points, Monitoring Points, and User Points. These subclasses are set up so that they can be converted to Junctions and used in the network, or so that they can be converted to network flags for analysis purposes. One reason for converting points into Junctions is that they can then be used as pour points for the delineation of watersheds from digital terrain models.

Simple points are used to model hydraulic structures that change the properties of flow by obstructing the river: dams, culverts, bridges, and the like. These can be thought of as valves in a pipe network. They may be completely

closed, partially closed, or completely open, and this status may block or restrict the flow in the river network.

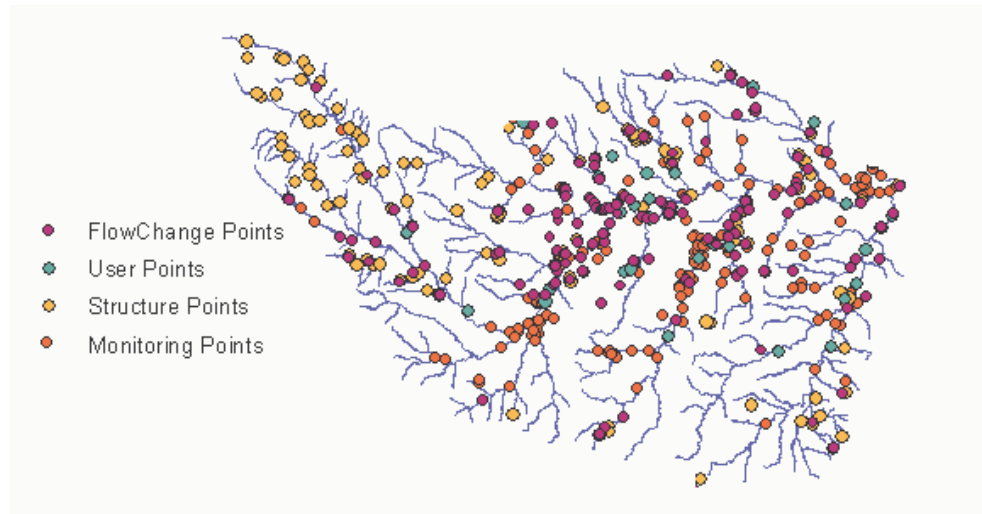


Figure 3.25 Hydro Points on the Lower West Fork of the Trinity River

Points are used to model most points of interest, such as Monitoring Points, Withdrawal Points, Structures and Discharge locations. The Water Points in the Washington and Oregon Hydrography Framework correspond to Hydro Points. These points can affect the volume of flow, so they can be used to perform traces of material moving along the network. Note that Hydro Point inherits from the base class Feature, and not from any network type (Edge or Junction). The Control Points can be displayed at the same time as the network features without interfering with the network. They are not required to be built into the network as Junctions, but through the use of spatial selection they can serve as query points for analyses.

Flow Change points are those at which something is added to or removed from the river; these points affect the mass balance of material in the river network. No change in the water flow occurs at monitoring points; they are locations at which observations are taken. Typical examples of Monitoring Points are USGS flow gage stations and Surface Water Quality Monitoring Stations. They are all referenced with the same addressing system, based on the LinearRef_ID and Measure properties.



Figure 3.26 Features that are represented by Structure Points (LCRA, 2000) and (Water CPI, 2000)

Structure points are intended to represent any feature, manmade or natural, which restricts or changes the movement of water. Examples include dams, bridges, weirs, and culverts. The Structure class represents hydraulic structures and other features that are in or on the network that change the hydraulic properties of the flow through the network by their presence. Typical examples of structures include detention ponds and culverts on small streams, dams and bridges on rivers, and weirs. These can also be natural features like waterfalls if they have significant effect on the hydraulic properties of the network. Typically

an irrigation turnout gate is not classified as a hydraulic structure, but as a Withdrawal under Flow Change Points. However, if the network of irrigation ditches were included in the network analysis, the turnout structure would be a Structure point and would be modeled as a valve on a pipe network.

Flow Change Points differ from Structures because at Flow Change Points, water is withdrawn from or discharged to the stream. Usually the corresponding flow data comes from permitting agencies that handle water rights and withdrawal permits, or from environmental agencies that permit discharges to the network. In any case, these points are significant to network analyses that deal with the mass balance of water and pollutants in the network and they are intended to be used as flags with a solver that tracks flow quantities down the network.



Figure 3.27 Features that are represented by Flow Change Points (Oregon Water Resources Department, 2000) and (Engineering Review, 1998)

Monitoring Points store the locations of gages that measure water quantity or quality. These can be linked to the Time Series data types, and are expected to

have temporal data associated with them for analysis purposes. Monitoring Points may also be subtyped or subclassed by the user. Monitoring Points include water quality monitoring stations, stream gage stations, rain gage stations, and any other type of fixed-location data collection points. These points can be tied to the Time Series Instant collected at their locations, through the identifier field Location_ID, allowing the display of gage data in graphical format, and comparison of gage readings at different locations. The Monitoring Points are well suited for subtyping, since most of them have similar attributes, but the attributes are not specified by the model because of the wide variety of attribute data expected within the water resources community.



Figure 3.28 Features that are represented by Monitoring Points (Conant Custom Brass, Inc., 1999) and (Institut für Meereskunde, 1997)

Hydro Points store Monitoring, Structure, and Flow Change Points—specific types of points used in water resources analyses. However, they also can

be used to store points for purely cartographic purposes. These are points that serve to enrich the content of maps, but are not used in analysis. Some examples of points used for cartography are locations of isolated (off-channel) ponds, rock outcrops, water wells, and small springs. Whatever data the user has in point form that does not fit into the model elsewhere goes into User Point. It is a good place to load data into so that the data can be organized and exported to other classes after schema application.

3.4.2.2 *Hydro Lines*

There are five basic types of hydrographic lines that participate in the network: natural streams and rivers, manmade canals or ditches, pipelines that carry water underground, connectors that are used when the original data had some obstruction covering the hydrologic feature, and artificial paths which represent the centerlines of lakes and other water bodies. These five types of lines comprise the drainage network in the NHD data model and in the City of Austin dataset. Stream network data from such sources has to be properly sorted and attributed before it can be loaded as Hydro Edge. Hydro Line can temporarily store network lines while they are edited in preparation for conversion to Hydro Edges.

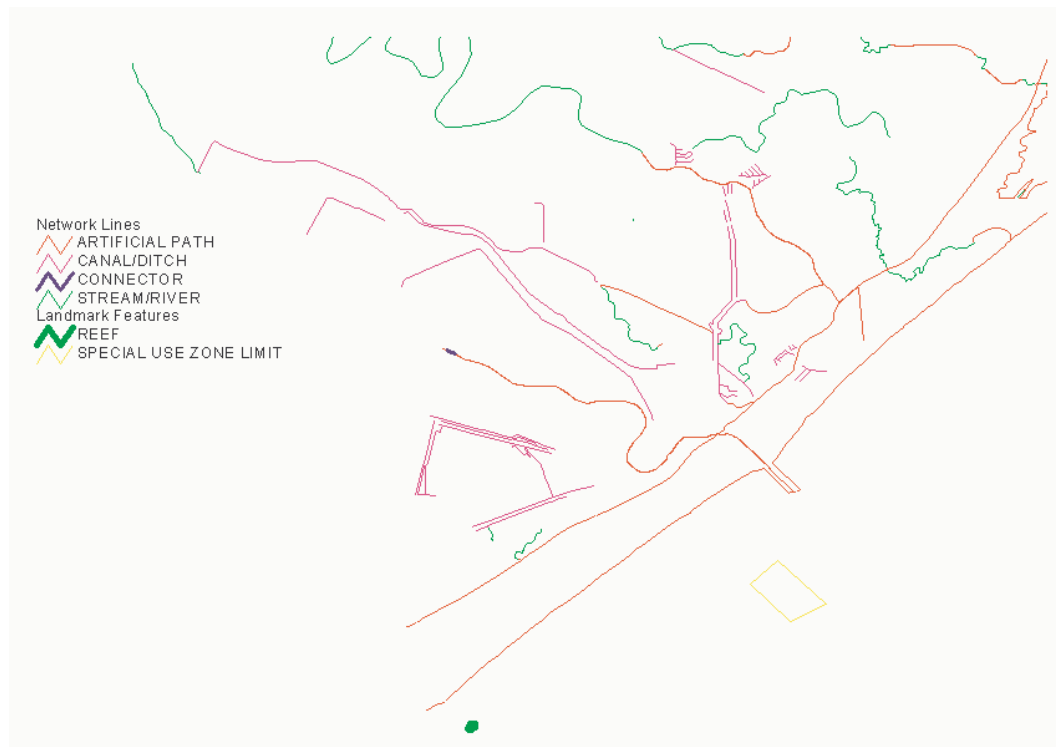


Figure 3.29 Hydro Lines taken from the NHD

There are many more types of hydrographic lines than those which participate in the network. Isolated ponds off the river network, shorelines, island boundaries, no-wake zones, swimming and recreation areas, roads, county and state boundary lines, jurisdictional boundaries for river authorities, and city limits are all marked off by lines which are important for cartography. They serve to provide a spatial reference for viewers of the data and so are necessary in the model. These types of lines are stored in the Hydro Features class Hydro Lines as subclasses or subtypes of Hydro Line. The user will specify new attributes for these classes to reflect important attributes of the data.

Closure lines for water bodies

Sometimes water bodies flow directly into each other, without a stream segment separating them. This is the case in the Highland Lakes area near Austin, Texas. The Colorado River there is dammed into a series of lakes that empty sequentially into each other. The dams are not part of the Hydro Edge system, but these water bodies are not connected, either. Regulatory and flow analysis purposes also demand that they be kept separate. Other examples of such situations include large bay systems along coasts which include side bays.

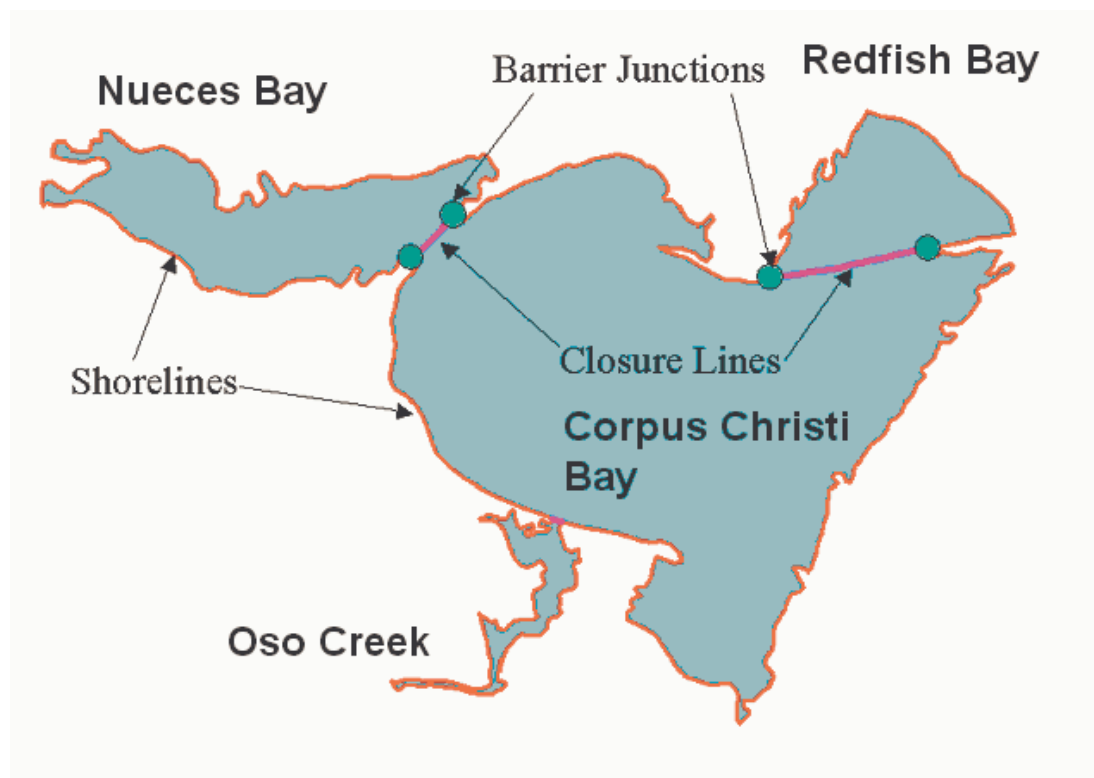


Figure 3.30 Closure Lines on Corpus Christi Bay

In Texas, at the outlet of the Nueces River, Nueces Bay does not empty into the Gulf of Mexico; it empties into Corpus Christi Bay. Redfish Bay and Oso

Creek also flow into Corpus Christi Bay, which then discharges to the Gulf. These are all distinct water bodies, but they are connected. For both cartographic and regulatory reasons, these water bodies need to be separated in the database. To accommodate such situations, closure line features were conceived. A closure line does not participate in the network and only serves to close off water bodies from each other.

3.4.2.3 *Hydro Areas*

Ordinary landmark areas have already been mentioned as types of lines stored in Hydro Line, but they may include a polygon representation that will be stored as Hydro Area. Examples of these data types are no-wake zones within water bodies, extents of counties or other jurisdictional areas, and inundation areas.

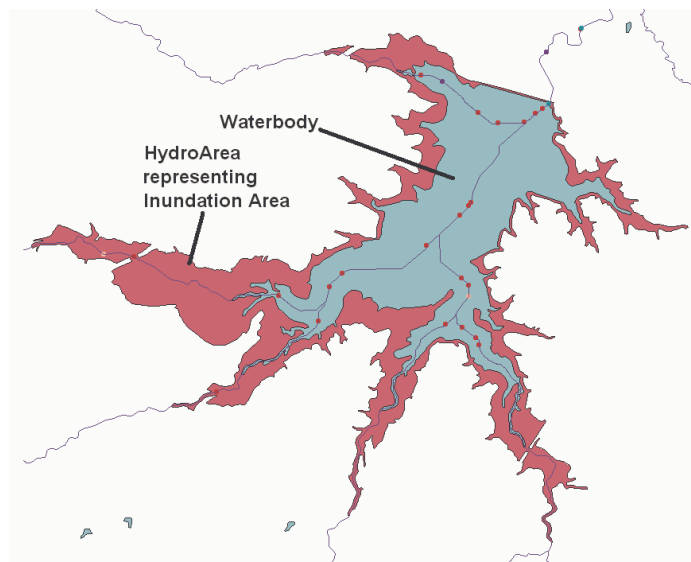


Figure 3.31 Hydro Area representing the Inundation Area of Benbrook Lake in the Trinity River Basin

Waterbodies can be stored in Hydro Area, particularly if they are not on the network. However, those water bodies that are on the channel of a river should be moved or copied into the Waterbody class. Land areas which participate in analysis, such as catchments and watersheds, land use maps and soil maps are stored in their respective classes as well, either Catchment, Watershed, or Hydro Response Unit. These classes are linked to the network through relationships so that the properties of these areas can be attached to the Hydro Network for analysis purposes.

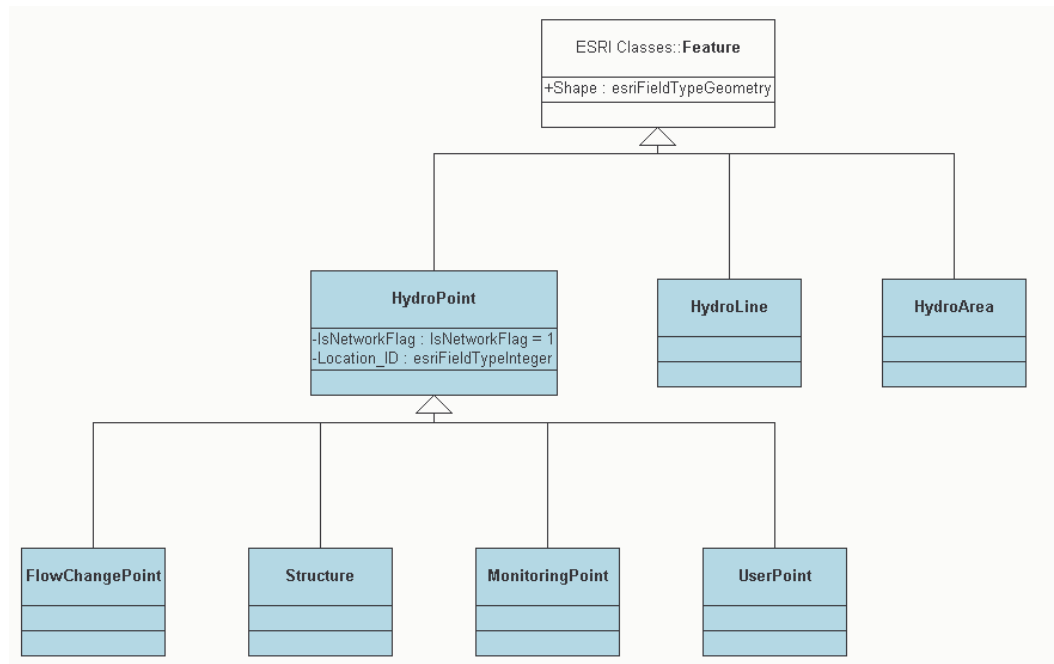


Figure 3.32 UML diagram of Hydro Features

3.4.2.4 Use of subtypes and subclasses

The subtypes Flow Edge, Virtual Flow Edge and Shoreline Edge inherit from Hydro Edge. This reflects their virtually identical properties, but their

different connectivity rules. Points of type Hydro Structure and Monitoring Point are subtypes of the class Hydro Point. This reflects their differing behavior and properties. Hydro Structures will likely be either subtyped or subclassed by the user, depending on the sorts of data they have to store.

In previous stages of the ArcGIS Hydro data model design, an attempt was made to create classes for many types of structures, but there were too many possible configurations. Some objects would only be used by a few people; others would be used by many but all would want them to have different attributes and behaviors. Rather than trying to specify every possible combination, the ArcGIS Hydro data model contains root classes which users can develop their own attributes for.

3.4.3 Channel Features

Channel Features are representations of the landscape in three dimensions. They describe the geometry of the river channel and its adjacent flood plains as a latticework of 3-D lines. They are traditionally built from cross-section surveys taken along streams. For this reason, the data structures used to describe channels are built of Cross Sections and Profile Lines.

Cross Section features are roughly transverse to the direction of flow in the river. Profile Lines are parallel to the flow in the river and come in five varieties. Left Bank, Right Bank, Thalweg, Left Flood Line, and Right Flood Line. The Thalweg feature delineates the centerline or deepest part of the river, depending on data availability. The centerline is not always in the same location as the deepest part of the channel; so, if the data is available the Thalweg marks

the deepest part of the channel; otherwise, the centerline is used for the Thalweg. Left and Right Banks indicate the main banks of the river identified in the cross-section data. The Left Flood Line and Right Flood Line can be positioned where the user desires, indicating the extent of the flood plain, the median path of water through the flood plain, or a major terrace.

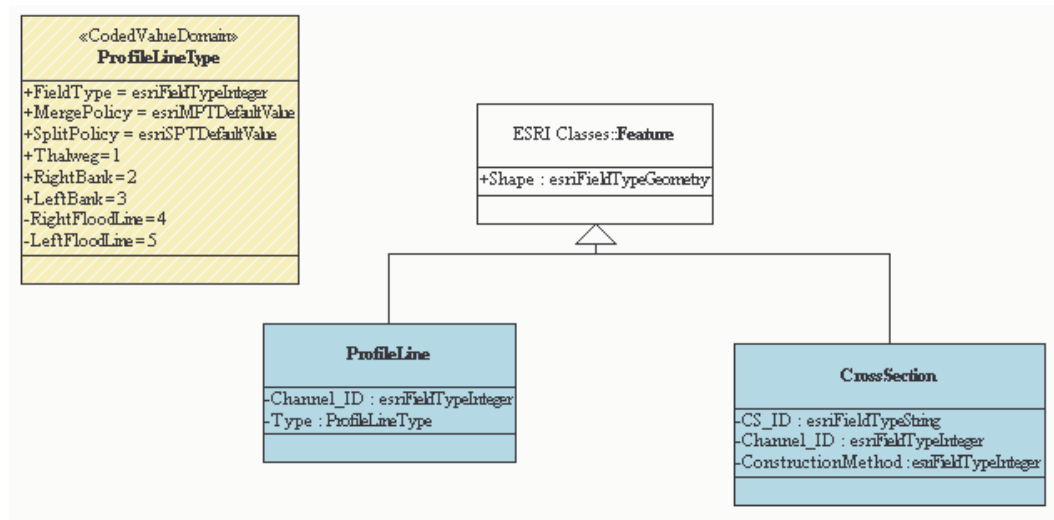


Figure 3.33 The UML diagram for Cross Section and Profile Line

The Profile Line class is not subtyped because the various types of Profile Line all have the same validation rules. Each subtype of a class may be assigned different domains for its attributes, this is one of the reasons for which subtypes are used. The other reason for subtyping is to enable the creation of connectivity rules within a network, but Profile and Cross Section Lines are not Network Features, so that does not apply.

The attributes of Profile Line are Channel_ID and Type. The Channel_ID links the five Profile Lines of a given channel model to one another and to the

Cross Section Lines that intersect them. Type is an integer code that indicates which type of Profile Line the feature is. The Type attribute on Profile Line is modified by a *Coded Value Domain* which restricts the data that may be entered in the Type field to one of the values specified in the domain. The text descriptions and their corresponding integer values are: Thalweg = 1, Right Bank = 2, Left Bank = 3, Right Flood Line = 4, and Left Flood Line = 5. When the attribute table for Profile Line is viewed, the Type field displays the integer values. When the Type field is edited manually, a drop-down list of text choices will prompt the user for a valid selection, which will be stored in the table as an integer value. Essentially, a Coded Value Domain is a built in look-up table that translates text into integers, providing the ease-of-use associated with text and the storage efficiency of integers.

Cross Section Line has the attributes CS_ID, Channel_ID, and ConstructionMethod. CS_ID is an identifier for the Cross Section that supports events located along the line. Channel_ID is an identifier that ties the Cross Section into a particular channel model. ConstructionMethod is an integer code that indicates the source of the cross section data, whether it is a GPS survey with 3-dimensional coordinates, or the result of a traditional survey.

Cross Section Events

In addition to the classes which make up the framework for the channel, there are two classes which provide auxiliary information about the channel. These are events similar to the Hydro Events found on the Hydro Network, but they are related only to the Channel Features. These two types of events are

derived from Object and are called CS Point Event and CS Line Event (for Cross Section Point and Line Events). The CS Point Event can be used to store information along a Cross Section identified by its CS_ID at a particular measure value. Typically, this information would include the roughness of the channel at that location, the vegetation height, or ecological data about the soil or biota. The CS Line Event stores the same type of data, but instead of recording the data at a discrete point location, it records the attributes of a strip of land along the profile line. CS Point and Line Events derive from the parent class Cross Section Event, which in turn derives from Object. Cross Section Event has the attribute CS_ID, which is used to indicate which Cross Section the event falls on. CS Point Event has the attribute Measure which records how far along the Cross Section the point is located. CS Line Event has the attributes FromMeasure and ToMeasure which record the starting and ending position along the Cross Section where the CS Line Event occurs.

3.4.4 Time Series

Time Series objects are currently under development, but the vision for these objects is that they will store tables of time varying data recorded at various locations along the river. The reason for storing this data is so that it can be meaningfully visualized, either in the traditional form of graphs or in some other way using map symbols, or both.

3.4.5 Relationships among objects

Object-to-object relationships are a means of passing notification when related objects are edited or deleted. They also enable the land surface parameters

to be tied to the stream network so that hydrologic analyses can use that data. Relationships are created and maintained through the use of key values in the attribute table. For related objects, at least one of the object classes has a unique identifier, and the other object class has an attribute value that lists the unique identifier of the related class. These fields are referred to as key fields, and the identifiers stored in them are called keys or key values.

Relationship of Edge Catchment to Flow Edge: Edge Catchments all drain to their corresponding Flow Edges. Flow Edge Catchments drain to Flow Edges, Virtual Catchments are sections of a Waterbody that are related to the Virtual Flow Edges flowing through it, and Shoreline Catchments drain to Shoreline Edges. These relationships are illustrated in Figure 3.34

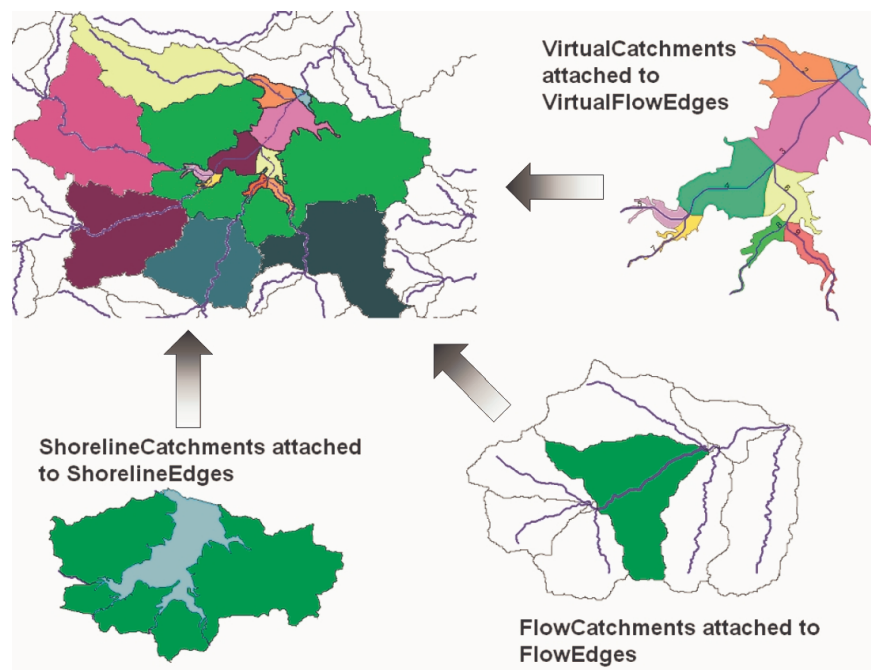


Figure 3.34 Illustration of the relationships between Edge Catchments and Hydro Edges

Relationship of Hydro Event to Hydro Edge: Hydro Events are located on Hydro Edges. They are linearly referenced, meaning that their coordinates are not specified in X and Y, but in terms of what Hydro Edge they fall on, and how far along that Hydro Edge they are located. The field LinearRef_ID is used to relate the event to the Hydro Edge. These can be either linear events or point events.

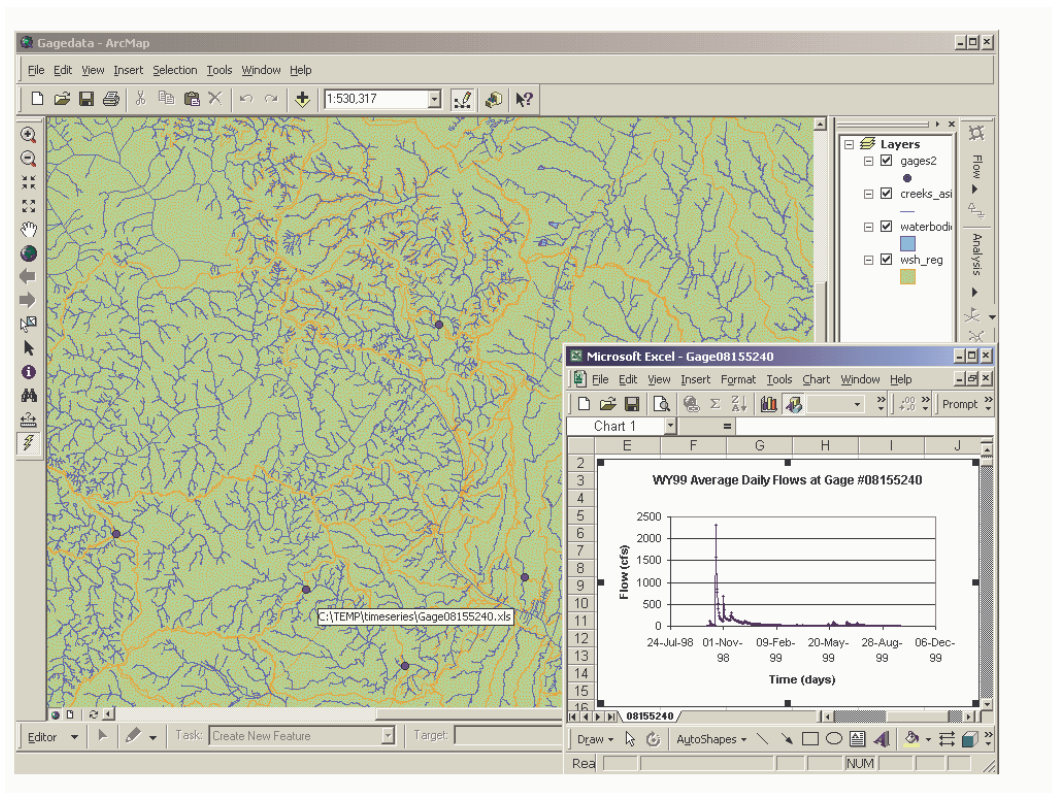


Figure 3.35 Relating points to time series of hydrologic data

Hydro Points are special Hydro Features in that they relate to Time Series objects. None of the other Hydro Features are designed with relationships. Time Series objects are assembled as tables of data with gage readings and other

observations made at that location. The Hydro Point class has the attribute Location_ID attached to it. This is intended to store the user-assigned identifier carried by that point location. The Location_ID appears on the Time Series Collection object, creating a relation in the database between the location and the data recordings taken there.

Points are related to time series of hydrologic data through the Location_ID field. The Time Series objects are intended to interface with many different types of databases, including DSS, Microsoft Access, and Oracle. Using relationships between Hydro Points and Time Series data, tables in these formats are imported into the geodatabase in a way that will allow the user to access and view the data trends.

Connectivity rules can be established for a network to enforce data integrity within the database. For example, Flow Edges can only connect to Virtual Flow Edges through a Shoreline Junction, because every time a flow line passes from a Flow Edge to a Virtual Flow Edge it has passed into a Waterbody. Shoreline Edges can only connect to other Shoreline Edges by means of a Shoreline Junction. Virtual Flow Edges must connect to each other at Virtual Junctions. These types of rules do not change connections digitized by the user, but incorrect connections are flagged for inspection when the user runs a database integrity check. Other database integrity tools, such as coded value domains, can be used to ensure attribute integrity just as connectivity rules enforce spatial integrity. In fact, connectivity rules are a special case of attribute domains that act on spatial data, instead of tabular data.

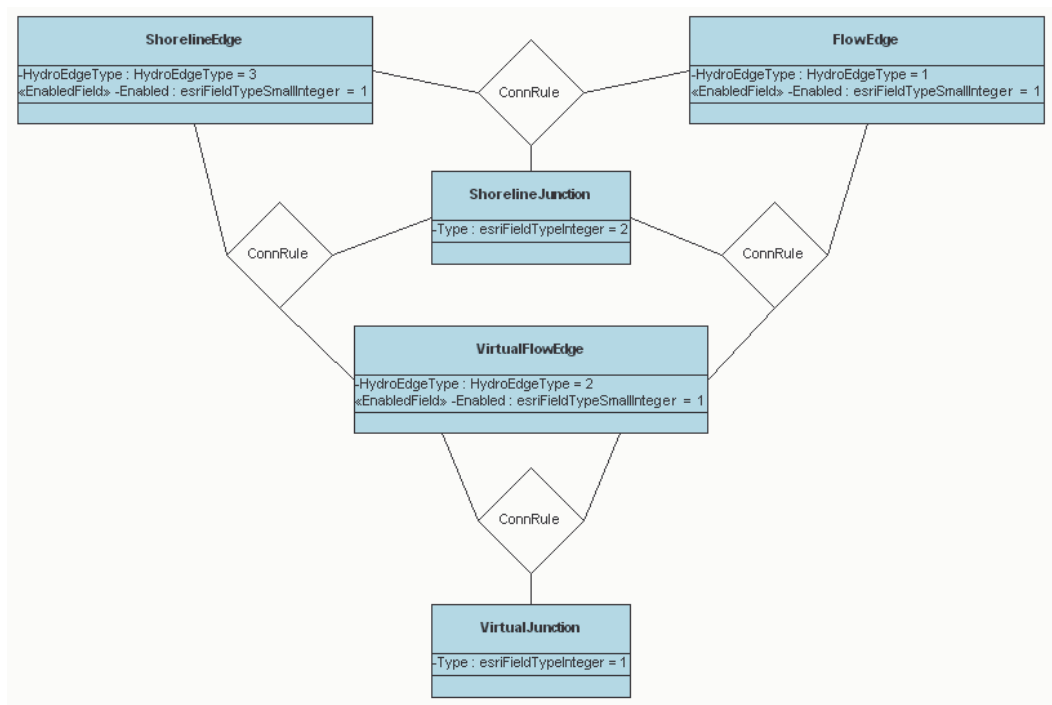


Figure 3.36 UML diagram of connectivity rules

Chapter 4: Procedure of Application

4.1 RECURSIVE MODEL DESIGN

Work on the model has been recursive and continuous since the beginning of the design process. The set of objects defined in the ArcGIS Hydro data model has been refined through continuous consultation with members of the GIS in Water Resources Consortium. Feature classes have been added, deleted, added back in, and modified repeatedly until the current state of the model was reached. The UML diagram representing the data model has been simplified as much as possible while still allowing for the display and modeling of the data necessary for hydrologic analysis. Simplicity is desired to comply with good modeling practice, but too simple a model requires more work to customize and does not provide a common structure which unifies different versions.

4.1.1 Summer Internship

Design of the prototype data model began during the author's internship at ESRI California in the Summer of 1999 with the author and a design team made up of ESRI and CRWR professionals deciding which objects to include in it. The real world objects the team sought to model were watersheds, river networks, channels, time series data, and landmark features. The initial stages were slow, and included the ramp-up time for the author to learn UML and the Visio Enterprise software. Once the knowledge was acquired, an initial skeletal model was drawn up in UML. After many correction phases and much debate, the result of the summer effort was a prototype data model. This was presented by the

author at the first consortium meeting in September of 1999 and was revised significantly by the author as a result of that meeting, particularly in regard to treatment of channel cross-section information.

4.1.2 Work with Consortium

The primary goal of the GIS in Water Resources Consortium is to create the ArcGIS Hydro data model. The driving concept behind the model is balance. It must follow good modeling practice for efficiency and elegance, but it must also be thorough and well-defined enough for users to be able to extend it with minimum effort. Following the September 1999 Consortium meeting were several meetings with individual Consortium members, including representatives of the NHD. Through consultation with all these groups a model which represented a good compromise between thorough specification of hydrologic concerns and a simple, elegant data structure was created. The Danish Hydraulic Institute (DHI) and the NHD assisted in many stages of model review and ensured that the ArcGIS Hydro data model will be compliant with their particular products and data formats.

Network elements were part of the ArcGIS Hydro data model design from the outset, but it quickly became apparent that polygons would have to be tied in represent drainage areas, and that time series data were also a crucial element. DHI was particularly instrumental in the design of the Time Series objects. The Army Corps of Engineers HEC programs were key to the initial designs of the Channel objects.

4.1.3 Work with ESRI

Initially the ArcGIS Hydro data model was based on Complex Edges and Complex Junctions. Further investigation of the uses to which the model will be put and the sources of data available prompted the use of Simple Junctions and Simple Edges in order to streamline the model and make it work more efficiently with ArcInfo 8. The decisions to simplify the objects were influenced by discussions with ESRI network architects, and result in improved efficiency and speed of analyses performed on the completed model. They also reduce the storage space used by the network, and simplify the tracing process by reducing the number of Junctions that must be considered.

4.2 IMPLEMENTATION IN VISIO

Visio software version 5, Enterprise Edition, is the CASE tool used to create the data model. This software was chosen because it is currently the only CASE tool that is fully COM-compliant and Repository-enabled. The diagrams are constructed in the software using a UML drawing template, which makes all the UML structures, such as classes, binary relationships, and inheritance relationships, available. These shapes are added to the diagram by dragging them from the template window. The base classes, such as Object, Feature, Simple Edge, and Simple Junction, are provided by ESRI with ArcInfo 8 in a Visio diagram file (.vsd). The ArcGIS Hydro data model was created by opening this file in Visio Enterprise, adding a new drawing page, and creating new classes on that page. All the properties of the classes, their names, attributes, default ranges and data types are specified in the Property Editor dialog box associated with each

class and relationship in the Visio drawing. Tagged Values are those data held internal to the object. By encapsulation, the user is not allowed to edit them in ArcGIS after the object has been created. However, these values must be initialized at some point, and using Visio is one way to do this. Tagged Values is the name of a tab within the Property Editor dialog. The geometry type of the feature (point, line, or polygon) and relationship keys can be set in the Tagged Values page of the Property Editor dialog.

The ArcGIS Hydro data model is divided into several drawing pages which reflect the logical framework of the model. They are located in Appendix A. The page divisions are inconsequential to the interpretation of the model by the computer or to its export to Repository format. The pages are all part of one drawing, just as a single building's blueprint may occupy several pages. The five pages used to depict the model are as follows: Network Features, Channel Features, Hydro Features, Time Series, and Connectivity Rules.

4.2.1 Export to Repository

Visio Enterprise is enabled to export UML Static Structure diagrams to the Microsoft Repository format. A Repository is a Microsoft Access database file (.mdb) that stores a collection of tables. These tables contain all the information that was graphically displayed in the UML diagram. The repository is used to generate stub code for creating the objects in Visual C++ and to create the schema in ArcInfo 8. Visio has a menu option for exporting the current diagram to Repository format. The user has only to provide the name of the Repository file and a title for the model then the translation process is entirely automated.

Additionally, the repository, as a single file ready to be read directly by the ESRI Code Generation Wizard and ArcCatalog, is an efficient way of transporting the model to users who have no need to edit it.

4.3 GENERATE CODE

In order to create geoobjects a developer has to write some code. Geoobjects are represented in the computer by a set of variables and instructions on what to do with them when certain events (such as a mouse click) occur. Most of the code required to create geoobjects is monotonous, simple, and repetitive. Thus, it is ideal for automation.

The Repository that is used to generate the schema in ArcGIS can also be used to create object code. ESRI provides a wizard that reads the repository and automatically creates the C++ code required to represent the objects from the UML static structure diagram. Within Visual C++ it is possible to add custom tools, and ESRI provides one such with ArcGIS called the CodeGenWiz. The CodeGenWiz is a wizard that runs in Microsoft Visual Studio and reads the Repository generated by Visio Enterprise. The wizard takes the information from the repository, which is a representation of the data model translated from the UML static structure diagram, and converts it into a C++ project with code stubs.

The stub code generated is comparable to a hollow building with all the plumbing and wiring in place to connect it to its environment. The code stubs created by the CodeGenWiz implement COM-compliant interfaces that allow the objects to interact with programs. Stub code is all that is required to define the basic objects from the UML diagram in ArcGIS or any other COM-compliant

program. However, if any unique or custom behaviors are to be implemented, they must be coded in by hand. Special behaviors include attribute domain rules, notifications to related classes, and shared geometry. Once any special code has been entered into the appropriate places in the stubs, the project is ready to be compiled.

4.3.1 .DLL File

When the Visual C++ project generated by the CodeGenWiz is compiled, the result is a Dynamic Link Library (.dll). The .dll file is a standard part of Windows architecture, and is a code library. Rather than including all the code needed to describe each object with the object itself in the GeoDatabase, .dll files store all the information in one place and the Geodatabase refers back to this .dll when doing something with the objects. A .dll has to be registered before it is recognized by the computer, but this is done automatically at the end of the compilation process. If the data model is to be used on a computer other than the one on which it was created, the .dll can be copied to other computers and registered to them. The .dll can be registered by dragging and dropping its icon onto the RegCat.exe icon located in the ESRI\ArcInfo\bin directory wherever ArcInfo is installed, as shown in Figure 4.1

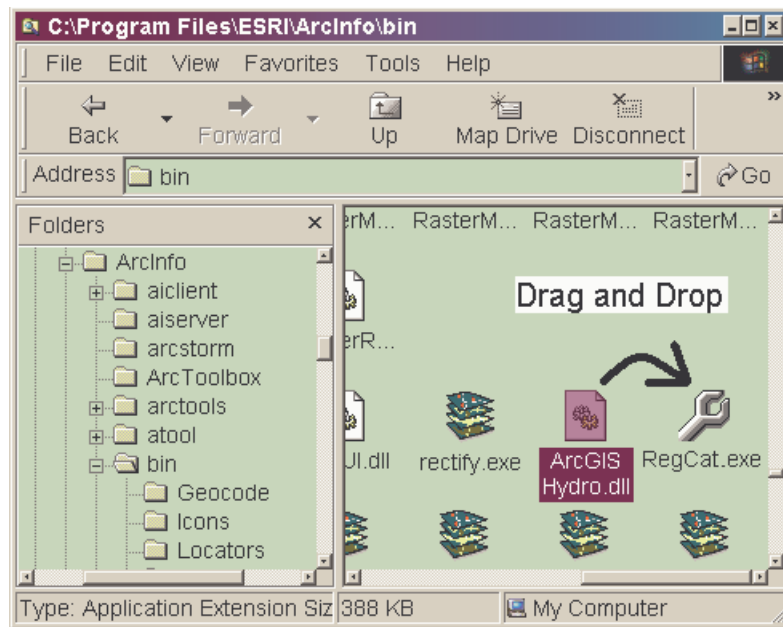


Figure 4.1 A .dll file about to be registered with RegCat.exe

4.4 USING THE ARCGIS HYDRO DATA MODEL

The steps presented up to this point are all part of the model design process. The addition of custom code to the geobjects is as much part of the design process as creating the UML diagram is. Once the Repository and the .dll file have been created, though, the model is complete and ready to be used. The remainder of this chapter explains how the model is used, and what customization may be done without returning to the CASE tools.

4.5 LOAD DATA

The first step in building a geodatabase with the ArcGIS Hydro data model is to find stream data. There are many sources of hydrographic data, on many spatial scales. The NHD is one example of a national-level data set

provided by the USGS. Data may also be available from local river authorities or state environmental agencies. Some data is also available on a global scale, such as the Digital Chart of the World or the ArcWorld data set available from ESRI. Alternatively, data may be generated from aerial photogrammetry, or digitized from paper maps. However it is acquired, what results is a set of lines representing streams and rivers, water body polygons, and some points along those lines which mark interesting locations. Most of these data are in file formats from previous versions of ESRI software, shapefiles or coverages.

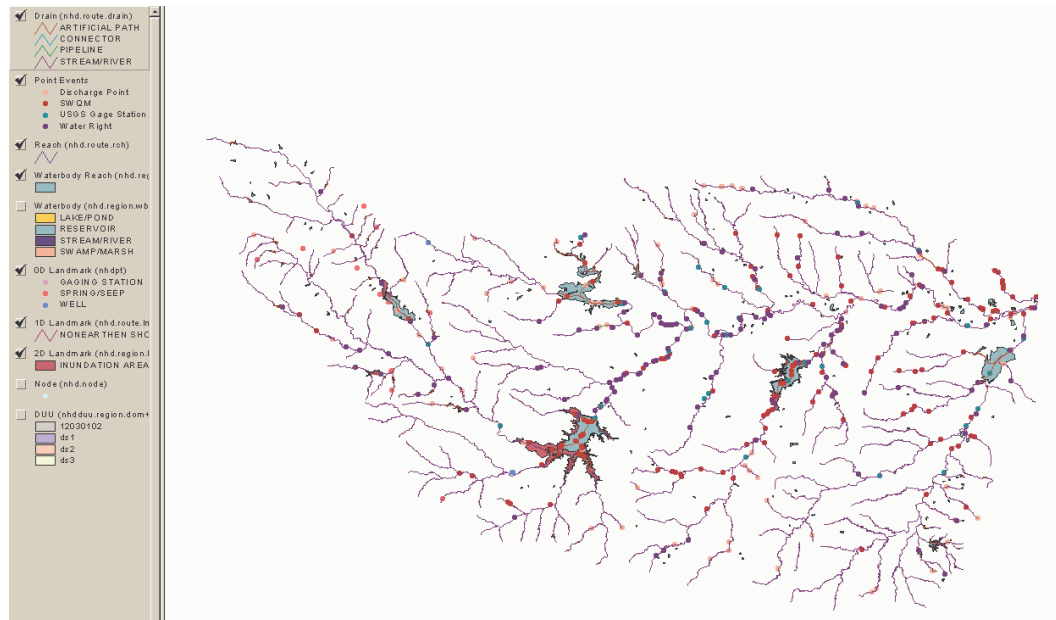


Figure 4.2 Sample of a dataset from the NHD for the Lower West Fork of the Trinity River

The geobjects are loaded into the geodatabase using tools provided with ArcInfo 8. A tool called the “Object Loader” is activated in ArcCatalog or ArcMap and existing data sets are loaded. The loader is a wizard which guides

the user through the steps of importing the data and then stores the data in the geodatabase. To use the Object Loader in ArcCatalog, right-click on the name of the target feature class and choose Load Data from the context menu. This version of the data loader is not as stable as the one that runs in ArcMap.

4.5.1 Adding the Object Loader to the Toolbar

In order to use the Object Loader with ArcMap, the tool must be loaded onto a toolbar. Choose Tools>Customize... from the menu, as shown in Figure 4.3.

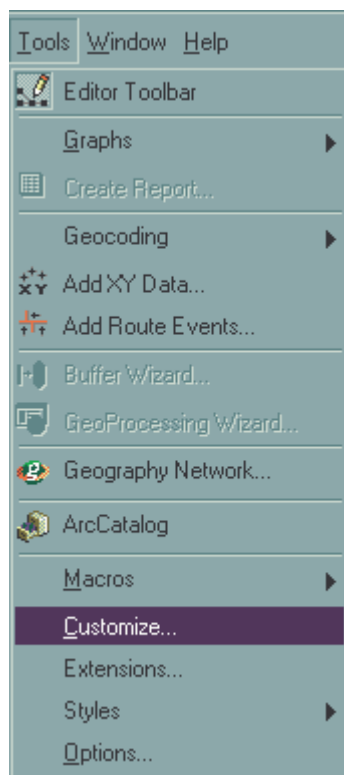


Figure 4.3 The ArcMap menu for customizing toolbars

In the resulting dialog box select the Commands tab, and then choose Data Conversion from the Categories list box on the left. This will display the corresponding list of Commands in the right-hand list box, as seen in Figure 4.4. Select Load objects in the Commands list box and drag the words to any toolbar on the screen. The Edit Toolbar is the most logical place to drop the Object Loader. Dropping the text onto the Edit Toolbar will place a button labeled Load Objects on the toolbar so it can be run.

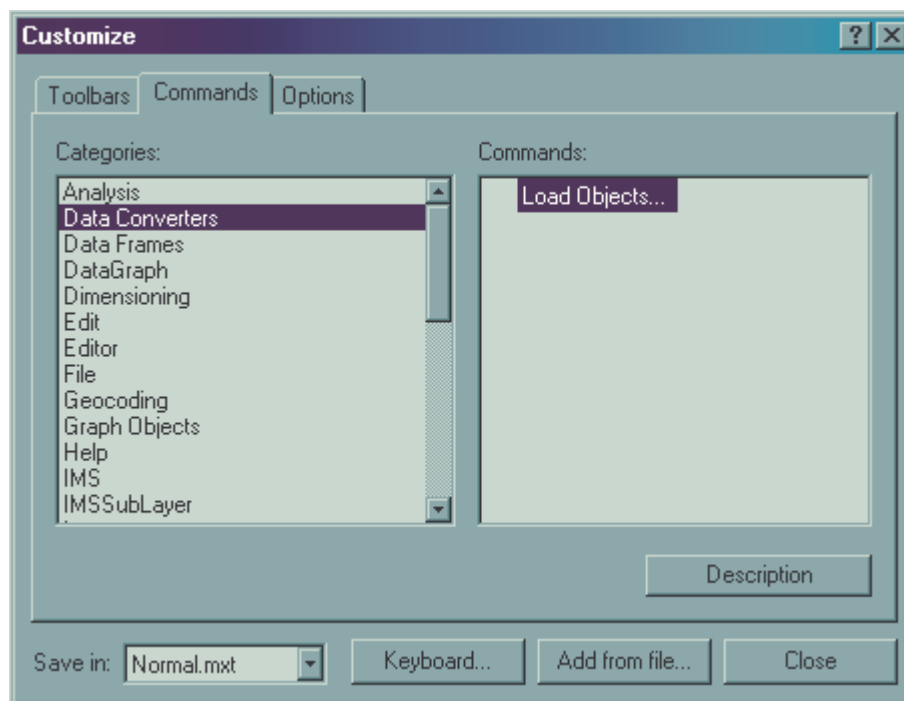


Figure 4.4 The toolbar customization dialog box in ArcMap

4.5.2 Use of ArcMap to load data

In order to load data, an edit session must be open in the directory the data will go into. If data from more than one geodatabase or directory is present in

ArcMap when the edit session is opened, the program will request that the user specify which directory or geodatabase to put the edit lock on. The edit lock must be placed on the target directory for data loading. To open the edit session, click Edit>Start Editing on the Edit Toolbar. With the edit session open, select the target feature class specified on the toolbar, then click the Load Objects button to run the wizard. Enter the name of the source dataset and edit the field names of the attributes to be loaded, if desired.

Once the data is loaded, the base cartographic data needs to be edited and simplified in order to make it useful. All the polygons representing isolated water bodies need to be separated from on-channel Waterbodies. The network lines must be clean, with all lines connected so there are no small gaps where lines are meant to intersect, and intersections must be clean, with no overhanging lines. Lines which represent simple streams and rivers are be assigned a type attribute matching the code for Flow Edges (HydroEdgeType = 1), and lines which represent the flow of water through water body polygons are assigned a type code for Virtual Flow Edges (HydroEdgeType = 2). If the water bodies in the cartographic dataset do not have centerlines within them, these must be digitized or otherwise generated (e.g., using the THIN function in the Grid module of Workstation ArcInfo).

Waterbodies that are on the stream network belong in one feature class. The lines that make up their borders should be copied out to the class that will correspond to the Hydro Edge and given a type attribute for the subtype Shoreline Edge (HydroEdgeType = 3). The LinearRef_ID on the Shoreline Edges is

matched up with the Waterbody_ID of the Waterbody from which the shorelines are derived.

Double-line rivers, those wide enough to be represented by a pair of bank lines on a map, can be represented as water bodies. If this is desired, the polygon shapes representing the river in the base cartography need to be transferred to the Waterbody class and their shorelines need to be converted to Shoreline Edges and connected to each other by Closure Lines to form a complete Waterbody polygon.

The coastlines of seas or oceans can also be converted to Shoreline Edges. Seas are often not represented by polygons on maps, so these will not be included in the Waterbodies to Shoreline Edges conversion. However, if the oceans are available as polygons, they can be used in the Waterbody class, and creation of shorelines will be taken care of automatically.

4.6 APPLY SCHEMA TO DATA IN ARCCATALOG

To prepare for applying the schema, data has to be appropriately edited and sorted into feature classes, and all collected into one feature dataset in a geodatabase. Classes that will have subtypes need to have an attribute field corresponding to the Type field in the schema. This field must be populated with the appropriate values before the schema is applied. After the features are sorted into feature classes and subclasses, the schema can be applied. Before application of the schema, data layers have no relationships or behavior. After the schema is applied, the feature classes retain the names as well as the attributes they had previously, but they are also assigned the attributes, relationships and behavior information of the feature classes which are mapped onto them. Any existing

attributes that correspond to the attributes from the schema will be renamed to match the schema, but other attributes will be unchanged. In contrast, if the data is loaded into an empty feature class created from the schema, existing attributes are stripped from the input data and only the matching attributes are carried on the resulting feature class.

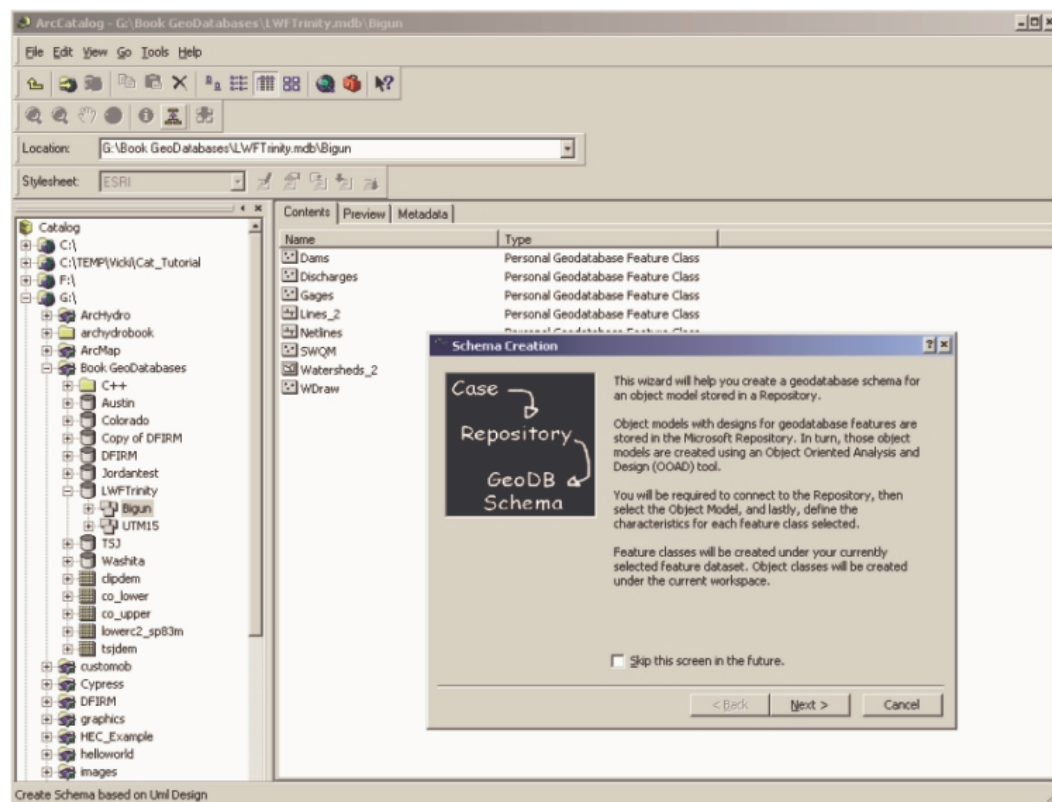


Figure 4.5 The opening dialog box of the Schema Creation Wizard

Data loaded into a geodatabase exists in whatever schema it was originally created with. Even if it has the same class and attribute names as the ArcGIS Hydro data model, it does not have the custom behaviors and relationships specified by the model until the ArcGIS Hydro data model schema is applied to it.

In order to create a complete instance of the ArcGIS Hydro data model in ArcInfo 8, the schema has to be applied. This converts existing data from simple point, line, and polygon classes to the custom objects specified by the .dll file generated by using the CodeGenWiz in Visual C++. Data that will go into network classes are loaded as simple features and converted from simple classes to network types after schema application.

4.6.1 Adding the Schema Generation Wizard to the Toolbar

The process of applying the schema is carried out through another wizard which is part of ArcCatalog. The tool must also be added to the toolbar before it can be used. To add the Schema Generation Wizard, choose Tools>Customize... from the ArcCatalog menu as shown in Figure 4.6.

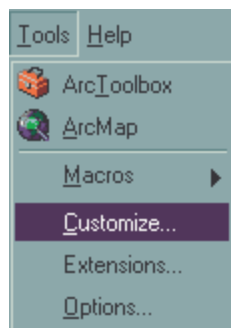


Figure 4.6 The ArcCatalog menu for customizing toolbars

In the resulting dialog box which is shown in Figure 4.6, click the Commands tab, and from the Categories list box on the left, choose Case Tools. This will cause Case Tools to appear in the Commands list box on the right. Drag the Case Tool from the Commands list and drop it on one of the toolbars in ArcCatalog.

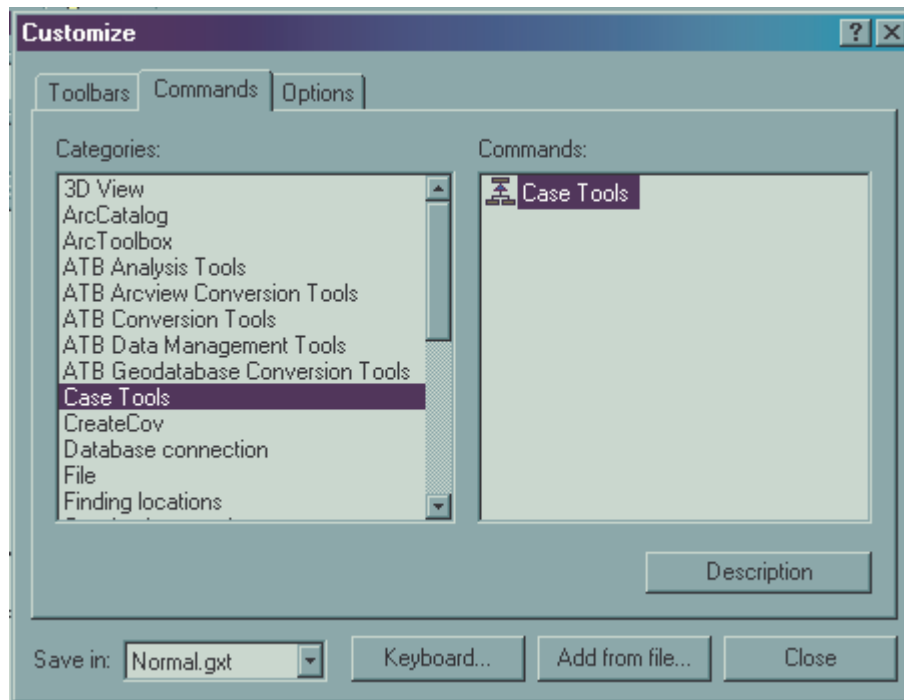


Figure 4.7 The toolbar customization dialog box in ArcCatalog

The Schema Generation Wizard can only be run when a feature dataset is selected in the data tree on the left side of the ArcCatalog window. Selecting a feature class or personal geodatabase from the tree, or selecting a feature dataset (or anything else) in the data preview pane on the right side of the ArcCatalog window will not activate the wizard.

4.7 GENERATE NETWORK

After the schema is applied, the network can be built. Network features support particular tools and solvers not supported by simple lines and points, and have restrictions placed on them in order to make that possible. For example, a Line feature need not be contiguous, and may be composed of a set of branching

or intersecting linear features. By contrast, an Edge must be a single line that is contiguous and does not branch, as in Figure 4.8. However, a simple Line cannot be part of a logical network and trace tasks cannot read it as they do Edges.

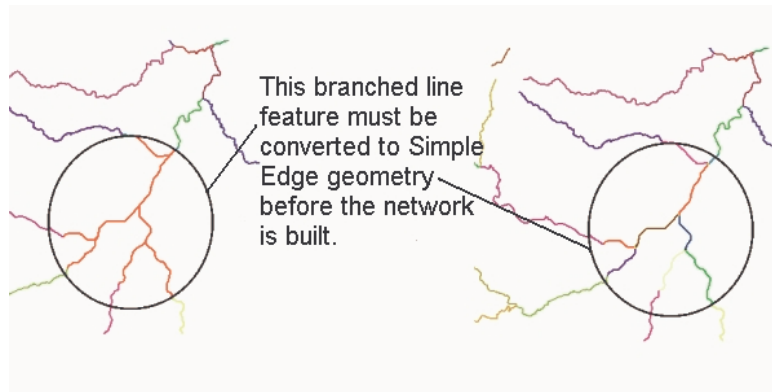


Figure 4.8 Preparing the data to generate a network

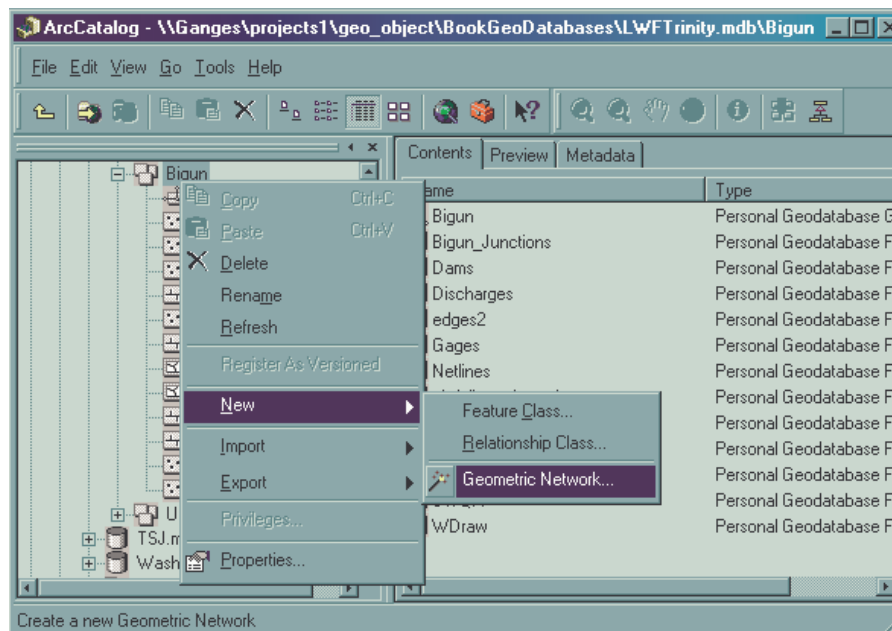


Figure 4.9 The Geometric Network wizard is invoked from ArcCatalog

The network is generated within a feature dataset using a wizard in ArcCatalog. There are options allowing the user to choose which feature classes will be used in the generation of the network connectivity tables, and whether or not the features involved need to be snapped in order to line up exactly. Unless the user has very specific precision requirements and must maintain the original point locations, it is recommended that the features be snapped, as show in Figure 4.10. Otherwise, it is likely that some of the Junctions will fall just a little bit off the network and will not be connected.

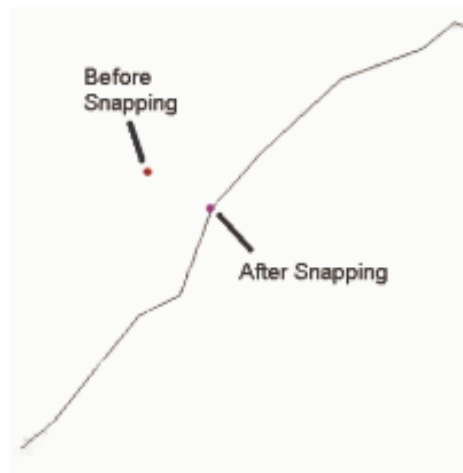


Figure 4.10 The results of snapping

The user may choose whether to adjust the locations of points to match the lines or vice versa, as well as being able to specify a tolerance distance beyond which a point will not be moved.

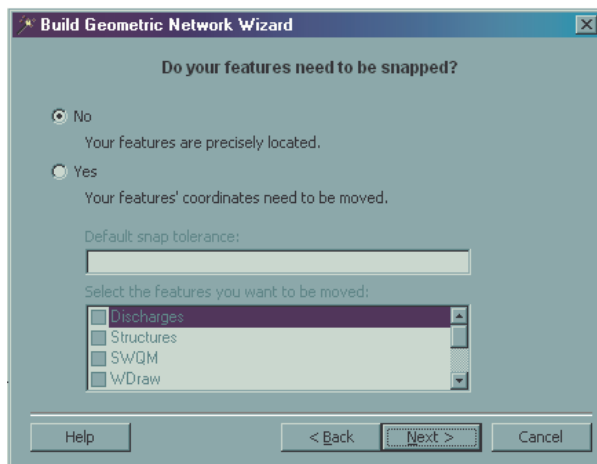


Figure 4.11 Snap Features dialog box from the Geometric Network Wizard

The next option presented in the Build Geometric Network Wizard is whether the Edges will be Simple or Complex in the network. For the ArcGIS Hydro data model, this must be set to create Simple Edges. Network creation also offers options regarding the existence of sources and sinks in the network. This option should be set to “yes”, and any point classes that will be used as outlets in the model should be checked off in the list of options.

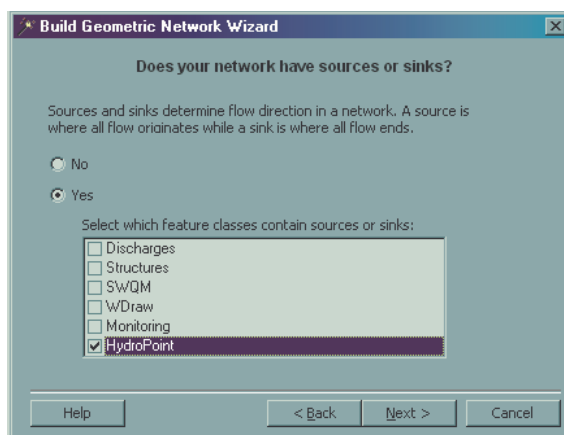


Figure 4.12 Dialog box for enabling Sources and Sinks in the Hydro Network

Sources and sinks are enabled, so that flow direction can be automatically determined through the network toolbar. Finally, the network wizard presents options regarding weights assigned to network elements. This can be set depending on the user's need. Trace solvers can be made to work with weights to determine least-cost paths.

Once the network is built, the flow direction can be assigned. This is performed in ArcMap. All the classes participating in the network, including the generic Junctions created by the network generation wizard, must be added to a map document in ArcMap. The Utility Network Analysis toolbar should be turned on, as well as the Editor toolbar. If there are no Junctions designated Sinks (AncillaryRole = 2), then at least one point needs to be created with this attribute. In the Property Inspector window, the AncillaryRole values are converted to their text equivalents None, Source, and Sink. In the attribute table, they are stored by their numeric values 0, 1, and 2.

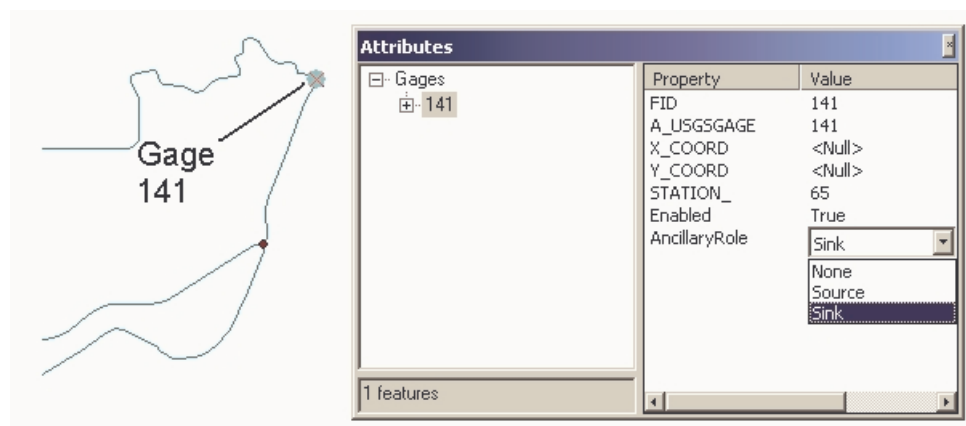


Figure 4.13 Property Inspector window showing the AncillaryRole list for a selected Junction

The points so designated will be read by the Network Analysis toolbar as Sinks in the network, places to which all flow attempts to direct itself. Once the Sinks are created, the flow direction can be assigned. An edit session must be open on the Editor toolbar for this to be performed. The FlowDirection button on the Network Analysis toolbar will run the solver for determining flow. This routine works very well on dendritic networks, but does not handle divergent or looping flow very well. If the network contains many loops or divergences, these need to be modified in some way to prevent the flow direction from being indeterminate. Once flow direction has been calculated, the arrows indicating flow direction are displayed. This allows quality control checks, shows where the flow was indeterminate, and identifies places where a flow direction was not assigned because the Edges were disconnected. The edit session may be closed at this point, with edits saved. Now directional traces can be performed on the network using the Utility Network Analysis Toolbar. Directional trace solvers can be allowed or forbidden to trace through zones of indeterminate flow in the Analysis>Properties menu option found on the Utility Network Analysis Toolbar.

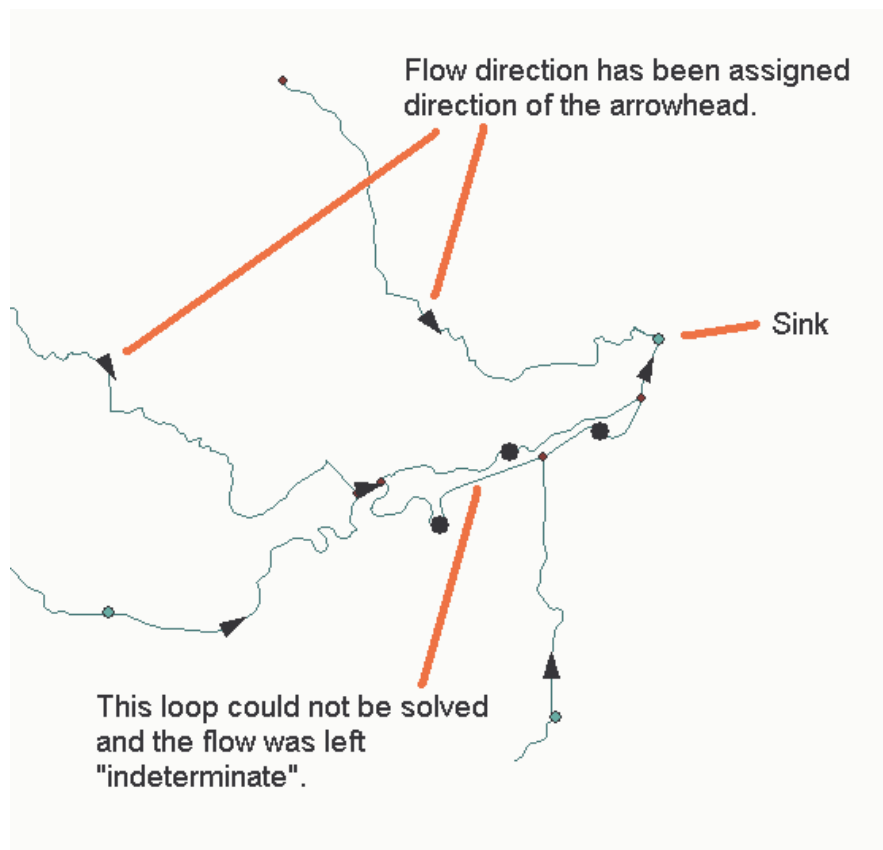


Figure 4.14 Using the flow direction information just recorded, measure values can be assigned to the Edges in the network in ArcInfo version 8.1.

Chapter 5: Results

The result of creating the model in ArcInfo is a geodatabase that bears the ArcGIS Hydro data model schema. The objects in this geodatabase have relationships and behaviors that facilitate creating input data to engineering models. As an example, the author has converted one Hydrologic Unit from the NHD into ArcGIS Hydro data model form. The Lower West Fork of the Trinity River (HUC 12030102) was provided by the NHD as a sample dataset.

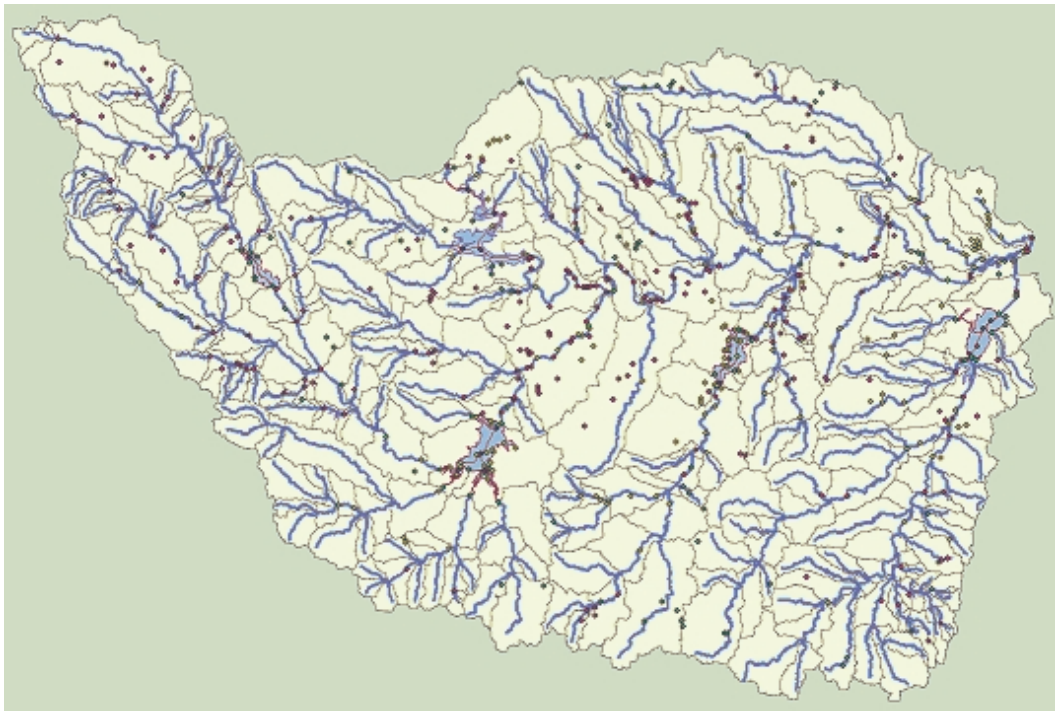


Figure 5.1 The Lower West Fork of the Trinity River in ArcGIS Hydro data model

In Figure 5.1, the Lower West Fork of the Trinity River is shown as it appears in geodatabase form in ArcMap. The Edge Catchments are in yellow, the

Flow Edges are blue, Waterbodies are blue polygons, Virtual Flow Edges are in red and Shoreline Edges are black. The points are from Hydro Junctions, and all of the Hydro Features subclasses.

The Hydro Edges in Figure 5.1 are taken from NHD flow validated data for HUC 12030102, the Lower West Fork of the Trinity River in Texas. The NHD data is in coverage format from ARC/INFO 7.2 and consists of three coverages as follows: NHD, NHDDUU, and NHDPT. Within the NHD coverage are the classes arc, node, polygon, label, and tic, as well as the subclasses region.lm, region.rch, region.wb, route.drain, route.lm, and route.rch. The region subclasses are aggregations of the polygon class, and the route subclasses are aggregations of the arc class. The extensions .lm, .rch, .wb, and .drain stand for landmark, reach, waterbody, and drainage network, respectively. The NHDDUU and NHDPT are auxiliary coverages and are not required for creation of the Hydro Network.

5.1 CONVERTING NHD INTO A HYDRO NETWORK

A feature dataset is created within a personal geodatabase to hold the new feature classes that are generated. Flow Edges and Virtual Flow Edges are constructed by first converting Route.rch from NHD into a coverage using the ROUTEARC command in Workstation ArcInfo. This done, ArcToolbox is used to run the BUILD tool and create topology for the arcs. This process breaks all branched routes, such as those found within waterbodies, into separate arcs. The newly-built line coverage containing the features from Route.rch is exported to geodatabase format and stored in a Line class called HydroLine. These features

will eventually become Flow Edges and Virtual Flow Edges. In ArcCatalog, a HydroEdgeType field is added to the HydroLine class to facilitate sorting the features into subclasses of Hydro Edge later.

Next, the region layer Region.wb is converted into a Polygon feature class called Waterbodies in the feature dataset. The waterbody centerlines are selected from HydroLine by using the Selection>Select by Location... menu option in ArcMap, and selecting the features from HydroLine that “are contained by” the features of Waterbodies. The selected HydroLines are assigned a HydroEdgeType = 2, denoting them as Virtual Flow Edges. The selection set of HydroLine is reversed so that all the reaches outside the waterbodies are selected and assigned HydroEdgeType = 1, meaning they are Flow Edges. Then Selection>Select by Location... is used again to select the set of features from region.wb that are intersected by the features of HydroLine. The selected set from region.wb is exported to a new coverage with the Region to Poly Coverage tool in ArcToolbox and BUILD is run with the line option to create a set of arcs. The new arc coverage is imported to Hydro Line using the Load Data tool in ArcMap. The newly loaded lines are assigned HydroEdgeType = 3, to indicate that they are Shoreline Edges.

Once the HydroLines are prepared, the Node class from NHD can be imported as a point class called Node in the feature dataset and given a new Type field. Nodes which are completely contained by the features of Waterbody are assigned a Type code of 1, for Virtual Junctions. Nodes that intersect a Shoreline Edge (HydroEdgeType = 3) are assigned a Type of 2 for Shoreline Junction. If

there are closure lines, any Nodes which intersect them can be assigned Type = 3 for Barrier Junction. These might have to be manually edited to determine which are Barrier Junctions and which are Shoreline Junctions. This series of steps will overwrite the Shoreline Junction in places where there is a Barrier Junction, which is not the ideal way to make these assignments but is preferred over manual editing. The most downstream node in the network, the outlet for the HUC, is assigned the Type code of 4, for Sink. The remainder of the Nodes are then deleted.

These steps have prepared the HydroLine and Node classes for conversion to Hydro Network classes. Any other NHD data which is useful or desirable to the user is loaded into the geodatabase at this time. Next, the schema is applied to the existing dataset. A network is created from the Hydro Edge and Hydro Junction types during schema application. If these classes are sufficient to define the network, then the prepared HydroLine and Node classes can be loaded into them using the ArcMap Load Objects tool. Otherwise, the empty network can be deleted and the network can be generated in the feature dataset including all the desired classes.

5.2 ANALYSIS CAPABILITIES

Some network analysis tasks, such as tracing through an entire network to determine which pieces are connected, require only a set of edges as input. The user chooses a start point and runs the solver; the results indicate all the edges in the network that are connected to that point.

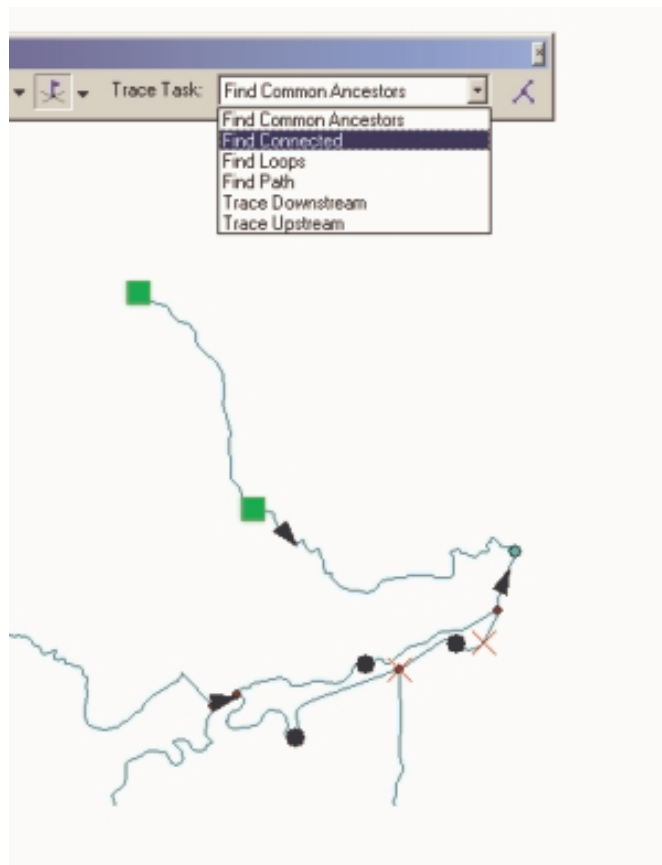


Figure 5.2 Trace solvers on the Utility Network Analysis toolbar

Other trace tasks, such as a point-to-point path tracing, require multiple point locations as input. These multiple locations are difficult to gather unless the user can place a graphic at the desired locations. In order to handle these situations, there are devices called Network Flags.

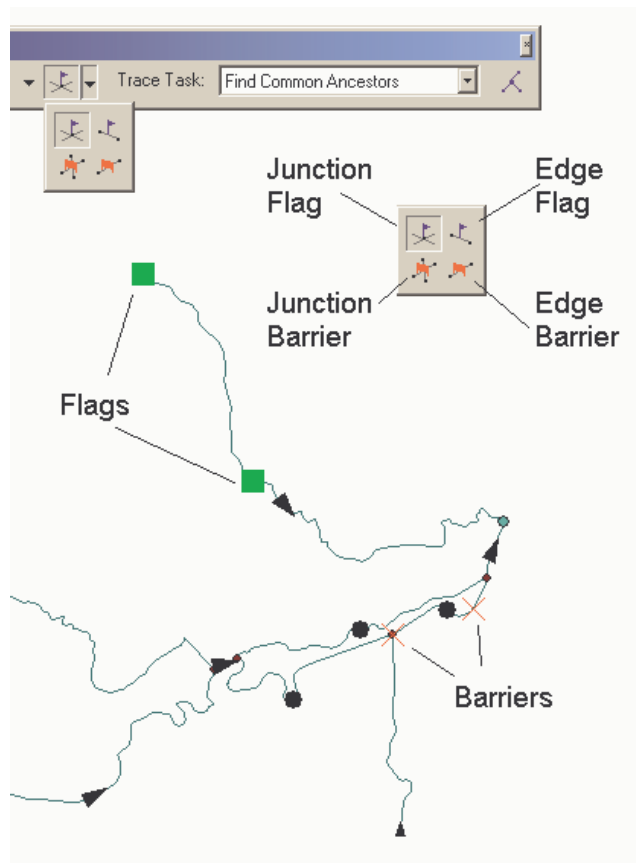


Figure 5.3 The four types of Network Flags

Network flags come in four varieties, Edge Flags, Junction Flags, Edge Barriers, and Junction Barriers. Flags are places on the network that have been tagged by the user as a point of interest. They are temporary and exist only until the user clears them out to perform a new analysis. Barriers are locations set by the user at which flow is blocked. No trace will navigate through them, so they allow users to isolate a portion of the network in which they are interested and ignore the rest.

The results of network traces are available in a variety of formats: as selected sets of features, as graphics drawn over the features, as the set of all features traced, or as the set of features that stopped the trace. The results can be stored and compared to one another and can be viewed independently of the network.

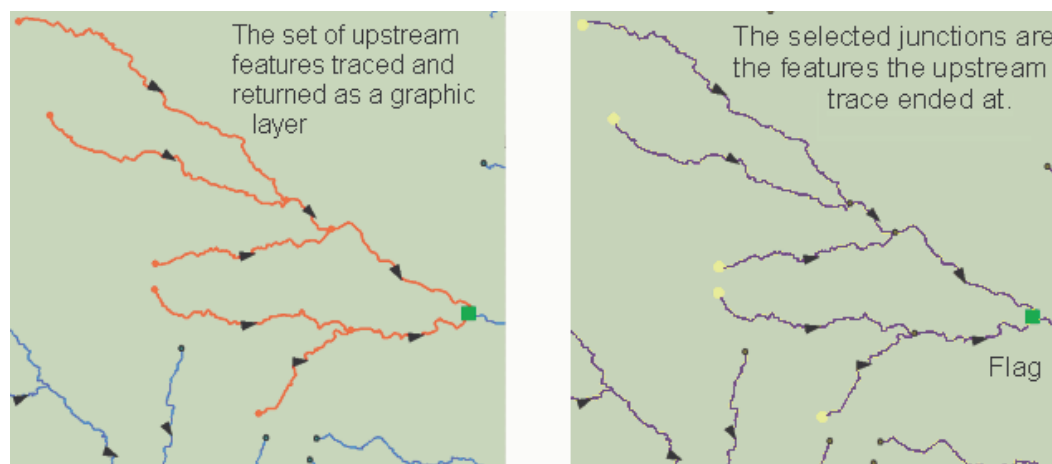


Figure 5.4 Different results configurations from the same upstream trace task

Solvers use weights assigned on the network when performing trace tasks. They can be used as in transportation analyses to find the “least cost” path, where cost is mileage or travel time between two points; or can be used as gates to indicate whether a particular element is open to be traced. They can also be used with customized trace tasks to track the flow of quantities through the network. This allows water, pollutants, or nutrients to be routed down a river and tracked as they go. The amounts from upstream can be accumulated at each junction, and the total contribution of all upstream areas can be determined at the Sink, or at various points of interest along the way.

Quantities being traced and accumulated down Flow Edges and Virtual Flow Edges will simply pass from edge to edge. The properties of the Edge Catchment, such as soil permeability, may be used to determine the quantity of water or chemical to be assigned to each Flow Edge. Some special considerations are required for getting flow from the land surface onto the Virtual Flow Edges. These flow quantities will be gathered in the Shoreline Catchments and passed to the Shoreline Edges. These will route flow in any direction along their length until the quantity is passed to a Virtual Flow Edge. From there, it is part of the flow network and will pass through the Virtual Junctions and other Virtual Flow Edges until it passes out of the Waterbody.

Chapter 6 Conclusions

The ArcGIS Hydro data model is the first user-created model produced to work with the new ArcGIS architecture. It has set a mark outside the water resources field for other GIS user groups to reach for. The full development of ESRI's ArcFM Water model took two years, so the ArcGIS Hydro effort is in good position for future growth by all accounts.

ArcGIS Hydro is the first data model in the domain that seeks to accurately reflect hydrography while using GIS analysis capabilities to actively support hydrologic and hydraulic analyses. The data model facilitates the development of data for engineering analysis, but it is not an engineering model. It can be used to bridge the gap between datasets created purely for feature inventory and engineering models that make use of hydrographic data. Use of the ArcGIS Hydro data model with data sources as varied as the NHD and HEC-HMS has shown that it is an extensible and customizable model.

There are an infinite number of ways to configure a schema for storing hydrographic data in a GIS. And yet, certain unifying characteristics have been observed by the author during this design process. The network of lines representing the river basin is the key to watershed-based analysis and is essential to hydrologic modeling. Use of time series and channel geometry data in conjunction with the stream network is what makes meaningful hydrologic analyses. Accordingly, the Hydro Network is the core of the ArcGIS Hydro data model. It is the frame or skeleton that all other data in the model is based on. The

terrain surrounding the network and the unique features within the landscape serve to enrich the model and make analysis more accurate. Consequently, the ArcGIS Hydro data model provides comprehensive modeling for hydrographic features and uses the power of ArcGIS to convert that into useful input to hydrologic models.

The design of the model was assisted by the insight gained from comparison with other data models. Studying how data modelers have previously resolved problems that faced the ArcGIS Hydro designers benefited the model as well as the author. Use of the network to anchor the data and watersheds to a contain the data is a common theme in all the models studied. As a result of the author's studies of linear referencing systems, the ArcGIS Hydro data model is very flexible with respect to addressing models.

Working with the design team members from the Consortium and ESRI also provided insights that could not have been gained otherwise in so short a span. The importance of semantics was driven home repeatedly, particularly in conversations with experts from other domains. Ultimately, determining what vocabulary to use in describing the components of the ArcGIS Hydro data model (and the model itself) were as important and weighty as designing the components.

The most significant lesson learned is that any model that can completely describe the features of the landscape is too bulky and unwieldy for use, some compromise is always required. The task of striking the balance between a fully specified hydrographic model and a clean, simple hydrologic model was the

classic problem of design. Learning to balance what is desirable with what is feasible was a tremendous part of the process, fully as significant as learning to read the UML or work with CASE tools. The result of a balanced design is a simple, elegant model that describes the most common features found in hydrologic analyses. The model does not just describe features, however, it takes a step further to support analyses by creating the input data they need using the capabilities of GIS.

Inclusion of the network functionality in ArcGIS is what has made the data model effort worthwhile. With the burgeoning amount of hydrographic data available, all of it striving to include a clean drainage network, now is a good time for a network-based GIS hydro model. Foremost among the concerns when drafting the model was ensuring that these existing datasets and industry standard models would be fully compatible with the ArcGIS Hydro data model. The functions available will automate and ease the task of data development. The available data will, at the same time, shape new models so that the two can work together in meaningful ways.

A great deal of advanced computer science technology was incorporated into the design of ArcInfo 8. Object-oriented programming and modeling, CASE tools, and COM are all part of what makes this project possible. Finding balance between good computer science and good hydrologic engineering was a major challenge in this project. However, a good balance has been struck, and the model allows for easy use and a reasonable degree of standardization as well as elegance and simplicity.

The bulk of the functionality incorporated into the ArcGIS Hydro data model is included by way of the Hydro Network. The Hydro Edges, Hydro Junctions, and Waterbodies are the central features in any hydrologic analysis, and it is fitting that they constitute the framework for analysis and hydrography in this data model. They support the use of linear referencing for location of events on the Hydro Network. Network analyses including directional tracing and material tracking can be performed with the logical network. Certain relationships and connectivity rules ensure that all data connections are made smoothly and that the Hydro Network functions as an integrated whole. Additionally, the Hydro Network is open and adaptable enough to incorporate data from widely varying data sources and to integrate with the other components of the ArcGIS Hydro data model.

6.1 FUTURE WORK

Most of the work required for the data modeling effort in this project has been completed. The model's content, semantics, and structure have been continuously refined, and the structure is deliberately basic so as to maximize its customization potential. Although the data modeling is largely complete, work remains to be done. The model is weak in regard to specification of behaviors for feature classes. The model attains its ultimate utility when a user takes ownership of it and completely suits it to their purposes. This may mean using ArcCatalog to add only a few descriptive attributes, or it may involve a full life cycle development of a model extension with new classes, behaviors, and relationships.

One interesting insight that has arisen from the development of the model is the value of relating watersheds to outlets on the network. Watersheds and catchments are ideal containers for data and their relationships to each other are best discovered and managed through the network. The ability to discover the whole landscape draining to a point of interest is inherent in the watershed to outlet relationship when the network is involved. A majority of the work involved in modeling runoff over the land's surface consists of discovering and measuring the properties of the land draining to a point of interest comprises. As this landscape-focused approach has been at the center of so much hydrology research in the past, it is certainly worth investigating the evolution of that relationship into the network domain.

Time Series data objects must be created for the model. The development of time series has been a troubled process. Much of the work performed on this task was unfocused and led to a confusing and unelegant design for the objects. Time series objects can be devastatingly simple, but they can also be sophisticated. Exciting new abilities to access time series data over the internet have made the drive to implement a Time Series object even more urgent. Balance, again, is the key to good design.

Component-object technology makes it possible to endow features with behavior for the first time in ESRI GIS and that is a capability which will be quite exciting when fully implemented. There is an understanding among the design team that certain rules will apply to some of the classes, and informal discussion of the desired behavior structure has been ongoing. However, a formal

delineation of the methods each object will support has not yet been done. At present, there are no behaviors specified, but in future work this must be a primary focus. The desired behaviors must be defined and the algorithms designed to carry it out. Once the code for implementing the behaviors has been added, the data model will be complete and ready for dissemination.

Appendix A

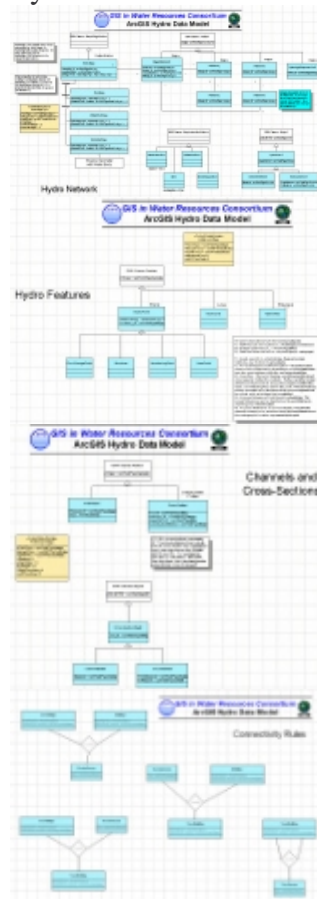
The UML static structure diagrams representing the ArcGIS Hydro data model are enclosed here. They are:

Hydro Network

Hydro Features

Channel Features

Connectivity Rules



N.B.: Time Series objects are currently under redesign and are not represented here.

Bibliography

- AGI (1996). The Association for Geographic Information (AGI) GIS dictionary.
<http://www.agi.org.uk/pag-es/dict-ion/dict-agi.htm>
- Booch, G., J. Rumbaugh, I. Jacobson (1999). The Unified Modeling Language User Guide. Addison-Wesley.
- Booch, G. (1991) Object-Oriented Design With Applications. Benjamin Cummings.
- British Columbia Ministry of Environment, Lands and Parks, Fisheries Branch (1996). Physical Data Model of the British Columbia Watershed Atlas. BC Ministry of Environment, Lands and Parks.
- Conant Custom Brass, Inc. (1999). Vermont Verdigris Rain Gauge.
<http://www.conantcustombrass.com/conant/vrg-2.html>
- DHI Software (2000). MIKE 11: A Modelling System for Rivers and Channels, User Guide. Vol. 1-2. DHI Water & Environment.
- Engineering Review (1998). "Water Rights/Water Management." Vol. 11, No. 1.
<http://www.engr.colostate.edu/college/adrg/review/oad.html> Colorado State University.
- Hathaway, R. (1996). Object-Orientation FAQ (COMP.OBJECT FAQ), Version: 1.0.9 4/2/96 Geodesic Systems, Inc. and Cyberdyne Systems Corporation.
<http://www.avalon.net/~wbachman/OOFAQ/oo-faq-toc.html>
- HEC (1999). HEC-GeoRAS, An Application for Support of HEC-RAS Using ArcInfo, Version 1.0. <http://www.wrc-hec.usace.army.mil/>
- HEC (1998). HEC-HMS, Hydrologic Modeling System, Version 1.0.
<http://www.wrc-hec.usace.army.mil/>
- Institut für Meereskunde (1997). Ship Rain Gauge.
<http://www.ifm.uni-kiel.de/me/research/Projekte/WOCE/ShipRainGage.html>
- LCRA (2000). Mansfield Dam <http://www.lcra.org/water/images/manflda.jpg>

- Maidment, D. (1999). Spatial Data Structure of the National Hydrography Dataset. <http://www.cwrw.utexas.edu/giswr/nhd/nhdprimer.pdf>
- Mitchell, A. (1999). The ESRI Guide to GIS Analysis, Volume 1: Geographic Patterns & Relationships. ESRI Press.
- NHD (1999). National Hydrography Dataset. <http://nhd.usgs.gov/>
- Oregon Water Resources Department (2000). Valve. <http://www.wrd.state.or.us/maps/wr-map.html>
- Rumbaugh J., et al. (1991). Object-Oriented Modeling and Design. Prentice Hall.
- Washington State Department of Ecology, Oregon and Washington Hydrography Framework Technical Work Groups (2000). Oregon and Washington State Framework, Clearinghouse Hydrography Data Dictionary, Physical Data Model, Version 1.0. Washington State Department of Ecology.
- Water CPI (2000). Self Regulating Tilting Weir and Fishway. <http://www.shogun.co.uk/watercpi/cpiphoto.htm>
- Webopedia (1999). <http://webopedia.internet.com/TERM/p/polymorphism.html>
- Zeiler, M. (1999). Modeling our World, the ESRI Approach to Geodatabase Design. ESRI Press.

Vita

Kimberley Marie Davis was born in Nacogdoches, Texas on January 17, 1975, the daughter of Sharon L. Davis and James H. Davis. After completing her work at Victoria High School, Victoria, Texas in 1992, she entered the US Air Force Academy in Colorado Springs, CO to study Middle Eastern History and Political Science. She transferred to Texas A&M University in College Station Texas in 1994 where she received the degree of Bachelor of Science in Civil Engineering in May, 1998. In August, 1998, she entered The Graduate School at The University of Texas at Austin.

Permanent address: 1701 E Polk Ave.
Victoria, TX 77901

This thesis was typed by the author.